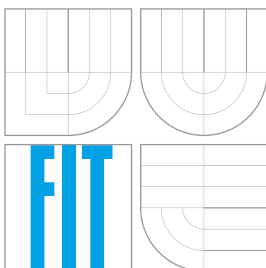


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

PŘEVOD BAREVNÝCH OBRÁZKŮ NA ČERNOBÍLÉ

COLOR-TO-GRAYSCALE CONVERSIONS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. PETR POSPÍŠIL

VEDOUcí PRÁCE
SUPERVISOR

Doc. Ing. MARTIN ČADÍK, Ph.D.

BRNO 2016

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií

Akademický rok 2015/2016

Zadání diplomové práce

Řešitel: **Pospíšil Petr, Bc.**

Obor: Počítačová grafika a multimédia

Téma: **Převod barevných obrázků na černobílé**

Color-to-Grayscale Conversions

Kategorie: Zpracování obrazu

Pokyny:

1. Seznamte se s problematikou konverze barevných obrázků na černobílé.
2. Vyberte a popište metody vhodné pro implementaci. Při výběru se soustředte na použitelnost výsledků z fotografického hlediska.
3. Do již existujícího systému implementujte alespoň tři zvolené metody.
4. S metodami experimentujte, posuďte jejich vlastnosti při převodu HDR obrazů, implementované algoritmy porovnejte a diskutujte možnosti budoucího vývoje.
5. Dosažené výsledky prezentujte formou videa, plakátu, článku, apod.

Literatura:

- Dle pokynů vedoucího.
- http://cadik.posvete.cz/color_to_gray_overview/

Při obhajobě semestrální části projektu je požadováno:

- Body 1 až 3 zadání.

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci dřívějších projektů (30 až 40% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Čadík Martin, doc. Ing., Ph.D., UPGM FIT VUT**

Datum zadání: 1. listopadu 2015

Datum odevzdání: 25. května 2016

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
602 00 Brno, Božetěchova 2



doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Černobílé obrázky nebyly nikdy nahrazeny těmi barevnými, a pravděpodobně také nikdy nebudou, protože jsou v mnohých oblastech nepostradatelné. V této práci byly zvoleny tři metody pro převod barevných obrázků na černobílé. Tyto metody byly následně implementovány. V rámci práce vznikla také testovací sada 24 HDR obrázků, která byla spolu s existující sadou 24 LDR obrázků použita pro otestování a posouzení vlastností implementovaných metod. Tyto metody byly také srovnány z hlediska přesnosti a preferencí pomocí dotazníku.

Abstract

Grayscale pictures were never replaced by color ones and will probably never be, because they are indispensable in many areas. In this thesis was chosen three methods for color-to-gray conversion. These methods was implemented afterwards. Within this thesis was also made test set of 24 HDR pictures, which was together with existing set of 24 LDR pictures used to test and evaluate features of implemented methods. These methods were also compared from the point of accuracy and preference using questionnaire.

Klíčová slova

převod barevných obrázků na černobílé, převod, černobílé obrázky, převod obrázků, HDR, fotografie, bilaterální filtr

Keywords

color-to-gray conversion, conversion, grayscale pictures, pictures conversion, HDR, photograph, bilateral filter

Citace

POSPÍŠIL, Petr. *Převod barevných obrázků na černobílé*. Brno, 2016. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Čadík Martin.

Převod barevných obrázků na černobílé

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Doc. Ing. Martina Čadíka, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Petr Pospíšil
24. května 2016

Poděkování

Tímto bych chtěl vyjádřit poděkování panu Doc. Ing. Martinu Čadíkovi, Ph.D. za trpělivost, věnovaný čas, poskytnutou testovací sadu obrázků, rady, konzultace a celkově vedení při řešení této práce.

© Petr Pospíšil, 2016.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod	3
2	Teorie	5
2.1	Barevné modely	5
2.1.1	RGB	6
2.1.2	CIE XYZ 1931	6
2.1.3	CIE L*a*b* 1976	6
2.1.4	LCH	7
2.1.5	CIE Luv 1976	7
2.2	Barevné prostory	9
2.3	Matematický aparát	11
2.3.1	Transpozice matice	11
2.3.2	Pseudoinverzní matice	11
2.3.3	Gradient	11
2.3.4	Laplaceův operátor	13
2.3.5	Poissonova rovnice	14
2.3.6	Helmholtz-Kohlrausch efekt	15
2.4	Histogram	16
2.5	Histogram matching	17
2.6	HDR	18
2.6.1	Pořizování HDR snímků	20
2.6.2	Kdy je HDR potřeba	23
2.6.3	Zobrazování HDR obsahu	24
2.7	Gaussův filtr	25
2.8	Bilaterální filtr	26
2.9	Chyby obrazu	28
2.9.1	Svatozář	28
2.9.2	Chyby způsobené kompresí JPEG	28
3	Návrh a implementace	30
3.1	Implementované metody	30
3.1.1	Robust Color-to-gray via Nonlinear Global Mapping	31
3.1.2	Efficient Color-to-Gray Conversion for Digital Images in Gradient Domain	37
3.1.3	Two-scale Tone Management for Photographic Look	42
3.2	Tone mapping studio	48
3.3	Implementace barevných modelů	49
3.4	Použité knihovny	50

3.4.1	matrix.h	50
3.4.2	poisson_pde	50
3.4.3	pfstools, pfstmo	51
3.4.4	boost::multi_array	51
3.4.5	FFTW	51
3.5	Použité vybavení	51
4	Testování	53
4.1	Testovací sada	53
4.1.1	LDR	54
4.1.2	HDR	54
4.1.3	Modely	55
4.1.4	Skript	55
4.1.5	Struktura souborů	56
4.2	Získávání HDR obrázků	56
4.3	Dotazník	57
4.3.1	Šíření dotazníku	59
5	Výsledky	61
5.1	Posouzení převedených obrázků	61
5.1.1	LDR	61
5.1.2	HDR	62
5.2	Data z dotazníku	64
6	Závěr	68
	Literatura	69
	Přílohy	72
	Seznam příloh	73
A	Obsah DVD	74
B	Plakát	75
C	Překlad a instalace programu	76
C.0.1	Závislosti	76
C.0.2	Překlad	76
D	Ovládání programu	77
E	Galerie převedených obrázků	78
E.0.3	LDR	78
E.0.4	HDR	78

Kapitola 1

Úvod

Nejstarší zhotovené fotografie byly černobílé. V té době bylo důvodem tohoto barevného podání technické omezení. Nejstarší fotoaparáty totiž pracovaly na principu osvětlování světlocitlivé vrstvy chloridu stříbrného. Tato metoda při použití jedné vrstvy reagovala pouze na množství dopadajícího světla a proto nemohla zaznamenat informaci o barvě.

I přesto, že barevné fotoaparáty existují již přes 100 let a jsou široce rozšířené a dostupné, černobílé obrázky mají stále svůj význam a půvab. Své uplatnění naleznou nejen ve světě profesionálního fotografování, ale také např. při černobílém tisku nebo ve zpracování obrazu.

Černobílé fotografie nebyly nikdy vytlačeny těmi barevnými, a pravděpodobně také nikdy nebudou. Výrobci fotoaparátů to vědí a proto téměř všechny dnešní moderní digitální fotoaparáty obsahují režim pro černobílé focení. V internetových galeriích je černobílá přibližně každá čtvrtá fotografie. I to je dokladem toho, že zabývat se převodem barevných obrázků na černobílé má smysl a budoucnost.

Hned po přistání prvních astronautů na Měsíci a odeslání fotografií na Zemi se objevily konspirační teorie, že se jedná o podvod, protože na fotografiích nebyly vidět hvězdy. Skutečné vysvětlení tohoto jevu je však jednodušší – použité fotoaparáty neměly dostatečný dynamický rozsah na to, aby na jedné fotografii zachytily současně Měsíční krajinu i hvězdy na obloze. Tento problém se pokouší řešit tzv. HDR fotografie, tedy fotografie s vysokým dynamickým rozsahem, které se v posledních letech staly velmi populární.

Existuje mnoho metod pro převod barevných obrázků na černobílé. Liší se nejen v kvalitě převodu, ale také například v rychlosti nebo celkovém způsobu své činnosti. Některé metody mohou být specificky zaměřeny na určitý typ vstupních obrázků. Cílem je, aby nové metody podávaly stále kvalitnější výsledky, a to i u snímků, které jsou svým vzhledem náročné pro převod na černobílé.

Toto téma jsem si zvolil proto, že se zajímám o počítačovou grafiku a zadání shledávám zajímavým, netradičním a s velkým potenciálem. V době volby zadání jsem toho o převodu obrázků na černobílé věděl velmi málo a chtěl jsem se o této problematice dozvědět více. I když se převod barevných obrázků na černobílé jeví na první pohled jako triviální, jedná se ve skutečnosti o celé odvětví počítačové grafiky a vývoj se zde posouvá rychle kupředu.

Cílem práce je porovnat a posoudit vlastnosti tří zvolených metod při převodu HDR obrázků. Tyto metody bylo třeba nejprve zvolit a popsat. Výběr je cílen na metody použitelné z fotografického hlediska. Součástí práce je také implementace těchto metod. Jedním z cílů je také s metodami experimentovat, srovnat je a diskutovat o možnostech budoucího vývoje. Posledním krokem práce je vytvoření prezentace dosažených výsledků formou videa, plakátu nebo článku.

Text práce je strukturován do šesti logicky členěných samostatných kapitol. Bezpro-

středně za úvodem následuje kapitola teorie (2). Tato kapitola obsahuje teoretický podklad pro dále popsané převodní metody, jsou zde popsány některé základní fotografické pojmy a také se zde poměrně široce pojednává o pojmu HDR. Dále následuje kapitola Návrh a implementace (3). Zde je popsán vytvořený program, včetně existujícího systému, do kterého byl zasazen. Podkapitola „Implementované modely“ 3.1 se pak věnuje samotným převodovým metodám. Obsahuje tři podkapitoly, jednu pro každou metodu, kde každá podkapitola obsahuje v první části popis principu metody a v druhé části samotný popis implementace. Další kapitolou je testování (4). Zde je čtenář seznámen s testovací sadou obrázků a postupem, jakým testování probíhalo. Je zde také popsán dotazník, který byl pro testování rovněž použit. V předposlední kapitole jsou shrnuty výsledky práce (5). V této kapitole jsou popsány rozdíly mezi jednotlivými metodami na testovací sadě obrázků a je zodpovězeno, jak je která metoda při převodu úspěšná. Také se zde nachází vyhodnocení dotazníku. Poslední kapitolou práce je závěr (6), který shrnuje základní fakta o práci a její výsledky.

Kapitola 2

Teorie

Tato kapitola poskytuje teoretický základ pro další text. Nejprve jsou zde popsány nejběžněji používané barevné modely. Volba popsaných modelů není náhodná, ale jsou to právě ty, se kterými a v kterých pracují metody pro převod obrázků. Další podkapitola popisuje použité matematické prostředky, zejména různé maticové operace. Závěr kapitoly se věnuje problematice HDR.

2.1 Barevné modely

Na úvod bude uvedeno několik základních faktů o barvách. Barvu nějakého předmětu si lze představit jako vlnovou délku světla, kterou povrch tohoto předmětu odráží. Barvu povrchu si pak lze představit jako jakýsi světelný filtr. Lidské oko vnímá viditelnou část elektromagnetického spektra o vlnové délce přibližně 380 až 770 nm [34]. Díky tomu dokáže zdravý člověk teoreticky rozeznat 10 000 barev [23]. Lidský vizuální systém není stejně citlivý na všechny barvy, nejcitlivější je na zelenou barvu (přibližně kolem 530 nm). Lidské oko je však schopné adaptace na velmi velký rozsah úrovní jasu.

Barevné modely představují různé způsoby jakými lze reprezentovat barvu v paměti počítače. Pracují tak, že kombinují barevné složky do výsledné barvy. Barevné modely lze rozdělit do několika kategorií, například podle typu používaných barevných složek. Velmi běžné je, že složkou je množství určité barvy ve výsledné barvě, například červené. Barevnou složkou ale může být také například sytost nebo jas.

Dále lze barevné modely rozdělit na aditivní a subtraktivní. Výchozí barvou aditivních modelů je černá. Přidáváním jednotlivých barevných složek se postupně zvyšuje jas výsledné barvy, až se z ní při maximální intenzitě všech složek stane barva bílá. Toto přesně odpovídá chování monitorů, kde se aditivní míchání používá. Vypnutý monitor – tedy ve výchozím stavu je černý a postupným přimícháváním barevných složek se zesvětluje. Naproti tomu subtraktivní míchání vychází z barvy bílé. S každým přidáním barevné složky se pak výsledná barva ztmavuje. Toto chování odpovídá tiskárnám, kde se používá právě subtraktivní míchání. Subtraktivní systém má však jeden problém. Kombinací všech barevných složek v jejich maximální intenzitě bychom teoreticky měli získat barvu černou. Při praktickém pokusu je však výsledná barva spíše tmavě hnědá nebo šedá. Řešením tohoto problému je přidání další speciální složky, která bude reprezentovat černou barvu.

Asi nejznámějším barevným modelem je barevný model RGB (2.1.1). Za své velmi široké rozšíření vděčí hlavně tomu, že je používán v digitálních fotoaparátech. Tento model je aditivní. Dalším velmi známým barevným modelem je CMYK. Tomu se však práce nebude

dále věnovat, protože se používá především při tisku, a pro samotný převod tedy není zajímavý. Tento model je subtraktivní.

Popsané barevné modely spadají mezi takzvané trichromatické kolorimetrické systémy. To znamená, že každá barva se skládá ze tří složek. Alternativou k tomuto přístupu může být reprezentace barvy pouze jako odkazu do tabulky barev. Taková tabulka barev se nazývá paleta. Jednou z používaných palet je paleta bezpečných barev pro použití na webu.

2.1.1 RGB

RGB [34] [23] je pravděpodobně nejznámější a nejrozšířenější barevný model. Používají ho jak fotoaparáty, tak například monitory nebo projektory. Zkratka RGB je odvozena z počátečních písmen barevných složek modelu. Jsou to: red (červená), green (zelená), blue (modrá). Model RGB tedy míchá tyto tři barevné složky, které jsou základními barvami. Všechny ostatní barvy jsou namíchány z těchto základních barev. Jak již bylo zmíněno, jedná se o aditivní model, takže vychází z černé barvy a přidáváním barevných složek stoupá jas výsledné barvy. Složky RGB lze vyjádřit na intervalu $\langle 0, 1 \rangle$, nebo častěji na celočíselném intervalu $\langle 0, 255 \rangle$. V takovém případě je potřeba pro uložení jedné barevné složky jeden bajt, a pro uložení celé jedné barvy pak 3 bajty. Z výše uvedeného vyplývá, že při použití celočíselného intervalu je možné v modelu RGB uložit $256^3 = 16\,777\,216$ různých barevných odstínů.

2.1.2 CIE XYZ 1931

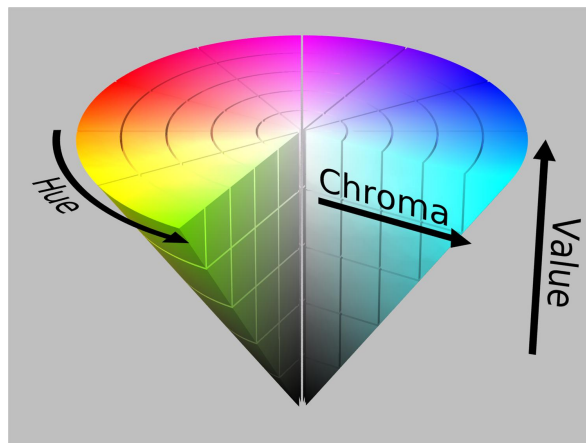
CIE XYZ je jedním z nejstarších definovaných barevných prostorů. Známý je hlavně proto, že tento prostor zavedl chromatický diagram, který se dodnes používá pro srovnání gamutů moderních barevných prostorů. Tvar diagramu odpovídá viditelné části elektromagnetického spektra. [34] Prostor XYZ předpokládá, že všechny zobrazitelné barvy je možné popsat jako vážený součet tří základních barev. Rozsahy složek XYZ jsou $\langle 0, 1 \rangle$. Z chromatického diagramu je však patrné, že ne všechny kombinace těchto složek tvoří validní barvy. Chromatický diagram CIE 1931 je možné vidět na obrázku 2.2. Tento diagram je založen na sérii experimentů. Lze jej popsat jako řez barevného prostoru při určité úrovni jasu. Cílem vytvoření tohoto diagramu byla možnost porovnávat barvy a pracovat s barvami bez omezení subjektivitou lidského vnímání barev.

2.1.3 CIE L*a*b* 1976

Tento model bývá pro zjednodušení často označován pouze Lab. Jeho výhodou je, že počítá s lidským vnímáním barev. Opět se jedná o třísložkový model. Jednotlivé složky mohou obvykle nabývat těchto hodnot: $\langle 0, 100 \rangle$ pro složku L a $\langle -128, 128 \rangle$ pro složky a^* a b^* . Složka L představuje světlost, zatímco zbývající dvě složky odstín barvy. Osa a^* ovlivňuje odstín v rozsahu (zelená, červená), zatímco osa b^* ovlivňuje odstín v rozsahu (modrá, žlutá) [34]. Tento model si tedy lze v 3D prostoru představit jako kouli, popř. jako „rozcestník“.

Oddělení světlosti do samostatného kanálu je důležitý a zásadní rozdíl oproti modelu RGB. Tento rys se vyskytuje také u mnoha dalších modelů. Díky tomu je možné snadno měnit jas bez ovlivnění barevného tónu obrázku. Toho je možné využít např. při tisku, když se nám vytištěný obrázek zdá příliš tmavý apod.

V důsledku této vlastnosti je tento model také vhodný k editaci fotografií. $L^*a^*b^*$ je podporován velmi rozšířenými editory GIMP a Adobe Photoshop. Umožňuje např. efektivně měnit tón snímku apod.



Obrázek 2.1: Znázornění prostoru složek modelu LCH¹.

2.1.4 LCH

Barevný model LCH podobně jako $L^*a^*b^*$ odděluje informaci o jasů od ostatních informací. Jeho složky L a C mohou nabývat hodnot $\langle 0, 100 \rangle$, složka H pak pracuje s rozsahem $\langle 0, 360 \rangle$. Význam těchto složek je: L – jas (lightness), C – sytost (chroma), H – odstín (hue). U modelu $L^*a^*b^*$ bylo možné odděleně ovlivňovat jas a odstín. U modelu LCH lze odděleně regulovat také sytost obrázku. Tento model si lze v 3D prostoru představit jako obrácený kužel, popř. válec. Tento kužel je směrem ke svému vrcholu tmavý a směrem ke své podstavě světlý. Dále je směrem ke svému středu bezbarvý a směrem ke svým okrajům obsahuje syté barvy. Po obvodu jeho podstavy se objevují všechny barvy. Viz obrázek 2.1.

Pro výsledný odstín je tedy rozhodující složka H. Tuto složku lze už kvůli jejímu rozsahu chápat jako natočení v kuželu. Hodnoty kolem 0° odpovídají červené, 90° žluté, 180° zelené a 270° modré.

U tohoto modelu se lze setkat také s označením HSB (hue, saturation, brightness), popř. HSV (hue, saturation, value). Model LCH také koresponduje s lidským vnímáním barev.

2.1.5 CIE Luv 1976

Luv je posledním z trojice modelů zmíněných v této práci, který odděluje jasovou složku od zbytku obrazové informace. Hodnoty složky L u všech těchto modelů si odpovídají. Při převodech mezi těmito modely se tedy složka L nemění.

Z uvedeného je zřejmé, že rozsah složky L je i u prostoru Luv $\langle 0, 100 \rangle$. Zásadně odlišné jsou však rozsahy složek u^* a v^* . U^* může nabývat hodnot $\langle -134, 224 \rangle$, v^* potom hodnot $\langle -140, 122 \rangle$. Lze se však setkat i s variantou, kde se používají rozsahy $\langle 0, 1 \rangle$. Z hlediska podobnosti má prostor Luv nejbližší k prostoru XYZ. To je způsobeno tím, že jak Luv, tak XYZ jsou ve skutečnosti barevné prostory, což bude upřesněno v kapitole 2.2. Chromatický diagram tohoto prostoru CIE 1976 je oproti CIE 1931 menší a barvy jsou v něm rozloženy rovnoměrněji [34]. Osa u^* v prostoru Luv odpovídá ose X v prostoru XYZ a osa v^* v prostoru Luv odpovídá ose Z v XYZ. Tento prostor vznikl z prostoru XYZ jako jeho jednodušší varianta.

¹Obrázek byl převzat z [//www.house-design-coffee.com/images/Lch-Cone.png](http://www.house-design-coffee.com/images/Lch-Cone.png)

Převody

Přesto, že prostory Luv a XYZ jsou si podobné, převody mezi nimi patří mezi ty komplikovanější. Převod [21] z XYZ do Luv se provede následovně:

$$L = \begin{cases} 116\sqrt[3]{y_r} - 16 & y_r > \epsilon \\ \kappa y_r & y_r \leq \epsilon \end{cases}$$

$$u = 13L(u' - u'_r)$$

$$v = 13L(v' - v'_r),$$

kde:

$$y_r = \frac{Y}{Y_r}$$

$$u' = \frac{4X}{X + 15Y + 3Z}$$

$$v' = \frac{9Y}{X + 15Y + 3Z}$$

$$u'_r = \frac{4X_r}{X_r + 15Y_r + 3Z_r}$$

$$v'_r = \frac{9Y_r}{X_r + 15Y_r + 3Z_r},$$

kde X_r , Y_r a Z_r jsou referenční hodnoty pro bílou barvu. Konstanty ϵ a κ jsou standardy definované CIE a jejich hodnoty jsou $\epsilon = 0,008856$ a $\kappa = 903,3$. CIE je zkratka slov Commission Internationale de l'Éclairage, což znamená Mezinárodní komise pro osvětlení.

Obrácený převod, tedy z Luv do XYZ se provádí podle těchto vzorců:

$$X = \frac{d - b}{a - c}$$

$$Y = \begin{cases} \left(\frac{L+16}{116}\right)^3 & L > \kappa\epsilon \\ \frac{L}{\kappa} & L \leq \kappa\epsilon \end{cases}$$

$$Z = Xa + b,$$

kde dílčí proměnné se vypočítají takto:

$$a = \frac{1}{3} \left(\frac{52L}{u + 13Lu_0} - 1 \right)$$

$$b = -5Y$$

$$c = -\frac{1}{3}$$

$$d = Y \left(\frac{39L}{v + 13Lv_0} - 5 \right)$$

$$u_0 = \frac{4X_r}{X_r + 15Y_r + 3Z_r}$$

$$v_0 = \frac{9Y_r}{X_r + 15Y_r + 3Z_r}$$

κ a ϵ mají stejné hodnoty jako v předchozím převodu.

Illuminant

Možností, jak zvolit referenční hodnoty pro bílou barvu, je několik. Tato volba závisí na tzv. *iluminantu*. Nejběžnější je tzv. standardní iluminant D65. Číslo 65 je odvozeno od teploty chromatičnosti 6500K. To znamená, že iluminant D65 používá osvětlení světlem takové barvy, kterou by mělo černé těleso zahřáté na teplotu přibližně 6500K. Tato barva je jednoduše řečeno „bílá“ a odpovídá jasnému dennímu světlu. Kromě tohoto iluminantu se ještě používají iluminanty D50, D55, nebo D75. Kromě iluminantů ze série D, existují ještě iluminanty A, B a C. A označuje osvětlení žárovkou, B se snaží simulovat přímé denní světlo a C simuluje také denní světlo, ale stíněné. Posledními typy iluminantů jsou iluminanty E, F2, F7 a F11.

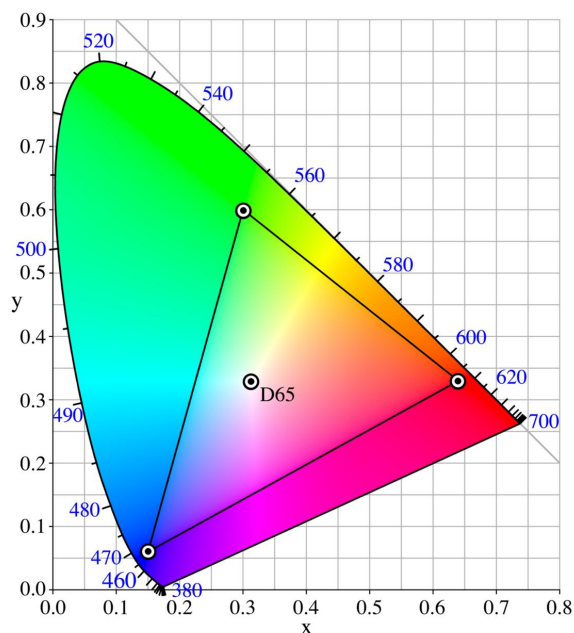
2.2 Barevné prostory

Barevný model RGB sám o sobě neurčuje interpretaci jednotlivých barevných složek. Pokud se k modelu RGB přidá definice chromatičnosti barevných složek, bude definován barevný prostor. To je tedy rozdíl mezi barevným modelem a barevným prostorem. Naneštěstí tyto pojmy bývají poměrně často zaměňovány. Barevný prostor přesně definuje množinu barev, kterou je možné v tomto prostoru zobrazit nebo snímat. Tato množina barev zobrazitelná v nějakém barevném prostoru se nazývá *barevný gamut*. Jednotlivé barevné prostory se odlišují právě svými gamuty. Gamut je popsán jako souřadnice všech barev daného prostoru na chromatickém diagramu CIE 1931. Součástí definice barevného prostoru také musí být hodnota bílého bodu. To jsou opět souřadnice na chromatickém diagramu odpovídající plně bílé barvě. Tyto souřadnice bílého bodu jsou potřebné pro převod do jiných barevných modelů. Posledním důležitým parametrem barevného prostoru je hodnota gamma [5]. Důležité je si uvědomit, že i když je možné zvětšovat barevný gamut, nezvyšuje se tím rozlišovací schopnost barevného prostoru RGB. Maximální počet zobrazitelných odstínů tedy zůstává stále stejný pro všechny barevné prostory nad stejným barevným modelem. Následuje výčet některých významných barevných prostorů založených na modelu RGB, jejichž vlastnosti jsou srovnány v tabulce 2.1.

- **sRGB.** Jedná se o pravděpodobně nejpoužívanější barevný prostor. Je používán v operačním systému Windows, ale je podporovaný také mezi open-source software. Používají ho také mnohé skenery nebo digitální kamery. Pro tuto práci je významný, protože tento barevný prostor se používá při implementaci převodů mezi barevnými modely. Gamut barevného prostoru sRGB je vzhledem k chromatickému diagramu 1931 znázorněn na obrázku 2.2.
- **Adobe RGB.** Adobe RGB je barevný prostor vyvinutý firmou Adobe Systems v roce 1998. Jeho gamut je výrazně větší než u prostoru sRGB, rozšířený je směrem do zelené barvy. Hodnoty červené a modré barvy jsou stejné jako u sRGB. Adobe RGB pokrývá zhruba polovinu chromatického diagramu CIE 1931.
- **Apple RGB.** Toto je barevný prostor vyvinutý firmou Apple pro CRT monitory.
- **Wide Gamut RGB.** Jak již název napovídá, tento barevný prostor opět rozšiřuje barevný gamut. Je v něm možné zobrazit přibližně 77 % chromatického diagramu CIE 1931. Stejně jako předchozí prostor byl vyvinut společností Adobe Systems. Díky

tomu, že tento prostor pokrývá tak velký gamut, je velmi vhodné používat na kanál více bitů než 8, tedy např. 16. V opačném případě se mohou objevit nežádoucí posterizační efekty.

- **Pro Photo RGB.** Tento barevný prostor přináší ještě větší gamut než Wide Gamut RGB. Jeho pokrytí chromatického diagramu dosahuje až 91 %. Byl vyvinut firmou Kodak v roce 2011. Barevný gamut tohoto prostoru je tak velký, že ve svých rozích vybočuje z chromatického diagramu CIE 1931. To znamená, že pomocí tohoto prostoru lze uložit neexistující barvy.



Obrázek 2.2: Gamut barevného prostoru sRGB vyznačený na chromatickém diagramu 1931. Modrá čísla po obvodu obrázku odpovídají vlnovým délkám příslušného odstínu.²

	γ	bílý bod		červená		zelená		modrá	
		x	y	x	y	x	y	x	y
sRGB	≈ 2.2	0.313	0.329	0.640	0.330	0.300	0.600	0.150	0.060
Adobe RGB	2.2	0.313	0.329	0.640	0.330	0.210	0.710	0.150	0.060
Wide Gamut	2.2	0.346	0.359	0.735	0.265	0.115	0.826	0.157	0.018
Pro Photo	1.8	0.346	0.359	0.735	0.265	0.160	0.840	0.037	0.000

Tabulka 2.1: Tabulka srovnávající parametry zmíněných barevných prostorů [21].

Kromě zmíněných prostorů založených na RGB existují také barevné prostory založené na modelu CMYK. Tyto prostory jsou vždy přímo navázané na konkrétní zařízení, na kterém se tento prostor používá.

²Obrázek byl převzat z [//upload.wikimedia.org/wikipedia/commons/a/a8/CIExy1931_sRGB.png](https://upload.wikimedia.org/wikipedia/commons/a/a8/CIExy1931_sRGB.png)

2.3 Matematický aparát

Tato kapitola obsahuje popis matematických aparátů použitých v dalších kapitolách. Základní a obecně známé matematické koncepty (např. matice) zde popsány nejsou.

2.3.1 Transpozice matice

Transpozice matice [20] je matematická operace, při které se jednoduše řečeno vymění řádky matice za sloupce. Pokud má tedy vstupní matice A počet řádků x a počet sloupců y , bude mít matice k ní transponovaná počet řádků y a počet řádků x . Transponovaná matice se značí A^T . Operace je jasně pochopitelná z následujícího příkladu.

$$A = \begin{pmatrix} 5 & 4 & 7 \\ 2 & 6 & 1 \end{pmatrix} \quad A^T = \begin{pmatrix} 5 & 2 \\ 4 & 6 \\ 7 & 1 \end{pmatrix}$$

2.3.2 Pseudoinverzní matice

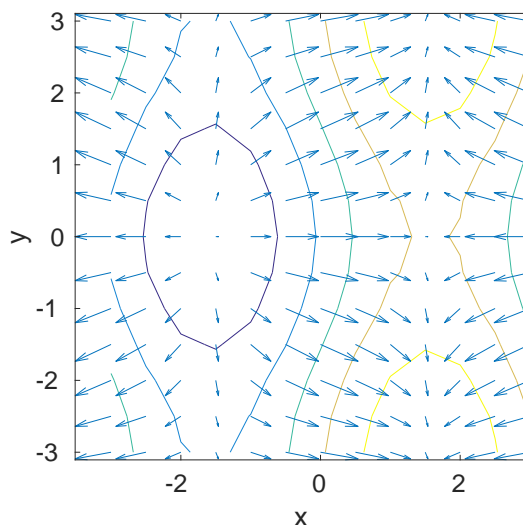
Nejprve bude vysvětleno, co je to inverzní matice. Je to taková matice, jejíž součin s původní (neinverzní) maticí se rovná jednotkové matici: $A \cdot A^{-1} = J$, kde A je původní matice, A^{-1} je inverzní matice k A a J je jednotková matice. U této operace funguje komutativita, takže platí i $A^{-1} \cdot A = J$. Pro ruční výpočet inverzní matice se často používá Gauss-Jordanova eliminační metoda. Při této metodě se vedle sebe napíšou vstupní matice a jednotková matice, a na obou těchto maticích se provádějí ty stejné ekvivalentní úpravy tak, aby z vstupní matice vznikla jednotková matice. Výsledkem pak je, že z jednotkové matice vznikne matice inverzní.

Aby tato (ale také jakákoliv jiná) metoda pro výpočet inverzní matice mohla fungovat, je nutné, aby vstupní matice byla čtvercová a regulární. To znamená, že musí platit $h(A) = n$, kde h je hodnost matice A a n je řád matice. Pokud toto neplatí, pak nelze inverzní matici nalézt, ale lze nalézt pseudoinverzní matici. Pseudoinverzní matice [1] je tedy „inverzní“ matice k matici, která není čtvercová. Pseudoinverze je zobecnění inverze a je důležité zmínit, že to není jednoznačně definovaná operace. Existuje více přístupů, jak pseudoinverzi provést. Nejznámější metodou je pravděpodobně Moore-Penroseova pseudoinverze, která se značí A^+ .

2.3.3 Gradient

Gradient [16] je vektor určující směr nejvyššího přírůstku. Takto definovaný gradient na skalárním poli tedy určuje směr růstu hodnot v tomto poli a je definován v každém jeho bodě. Gradient lze tedy chápat také jako jakýsi prostředek pro převod skalárního pole na vektorové pole, protože výsledkem takového převodu bude právě vektorové pole. Příklad grafického znázornění takového vektorového pole je na obrázku 2.3. Jedná se o kartézský souřadnicový systém v rozsahu $[-5; 5]$ v obou směrech s gradientem vykreslovaném po kroku 1. Délky vektorů odpovídají strmosti funkce. Zdrojové skalární pole obsahovalo graf funkce $z = \sin(x) + \cos(y)^2$. Na tomto obrázku jsou zobrazeny také vrstevnice. To jsou čáry procházející místy, kde je gradient konstantní. Je také vidět, že gradient je na tyto vrstevnice vždy kolmý.

V matematice se gradient označuje symbolem nabla ∇ a je složený z parciálních derivací všech vstupních proměnných. Při zpracování obrazu se nejčastěji používá gradient



Obrázek 2.3: Ukázka gradientního pole s vyznačenými vrstevnicemi.

dvourozměrného pole, který je definován takto:

$$\text{grad} f = \frac{\partial f}{\partial x} \vec{i} + \frac{\partial f}{\partial y} \vec{j}, \quad (2.1)$$

kde \vec{i} a \vec{j} jsou vektory ve směru os x a y . Používá se také zápis zachovávající jednotlivé parciální složky:

$$\text{grad} f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right) = \nabla f \quad (2.2)$$

Jak je vidět v rovnici 2.2, toto zároveň přímo odpovídá definici operátoru nabla. Nabla však nemusí označovat pouze gradient, je to pouze pomůcka pro zkrácený matematický zápis. Kromě gradientu se používá také např. rotaci nebo divergenci. Gradient nemá žádnou jednotku a lze s ním počítat v libovolném souřadnicovém systému, kromě kartézského tedy také např. ve sférických souřadnicích nebo v souřadnicích válcových.

Obecně je gradient ve zpracování obrazu velmi užitečný nástroj, protože spousta obrazových technik pracuje v gradientní doméně. Namísto počítání se samotnou hodnotou bodu se tedy počítá rozdílem (s podobností) se sousedními body. Velmi výhodné je použití gradientní domény např. pro „bezešvé“ mixování více obrázků dohromady (tvorbu koláží), které je za použití klasického přístupu náročné a použití gradientního přístupu přináší velmi dobré výsledky. Obvykle se zpracování obrázku v gradientní doméně skládá ze třech kroků:

1. Konstrukce gradientního pole
2. Grafické úpravy, např. aplikace filtru, deformace, ...
3. Rekonstrukce původního obrázku

Už samotná tvorba gradientního pole není triviálním problémem, protože existuje více metod, jak takové gradientní pole získat [31]. Mezi nejčastěji používané patří: dopředná difference 2.3, zpětná difference 2.4 a centrální difference 2.5:

$$f'(x_i) = y'_i \approx \frac{y_{i+1} - y_i}{x_{i+1} - x_i} \quad (2.3)$$

$$f'(x_i) = y'_i \approx \frac{y_i - y_{i-1}}{x_i - x_{i-1}} \quad (2.4)$$

$$f'(x_i) = y'_i \approx \frac{y_{i+1} - y_{i-1}}{2h}, \quad (2.5)$$

kde h je velikost použitého kroku a proměnná y je závislá na proměnné x . Rekonstrukce původního obrázku je ještě náročnější. Teoreticky by bylo možné použít metodu prosté integrace, a rekonstruovaný obrázek by se pak měl od originálu lišit pouze o integrační konstantu. Při použití prosté integrace by ale mohlo dojít ke vzniku chyb v obraze. K tomu zpravidla dochází, pokud gradientní pole není konzistentní. A právě při provádění grafických transformací se často konzistentnost gradientního pole poruší. V takovém případě je potřeba řešit Poissonovu rovnici, která nekonzistentní pole umí vyřešit. Více o této rovnici a o možnostech řešení je popsáno v kapitole 2.3.5.

Obrázek 2.3 prezentuje obecné, matematické pojetí gradientu. Na obrázku 2.4 je ale vidět, jak vypadá gradient pro běžnou fotografii, vykreslený vedle ní ve stejném měřítku. Světlejší místa znamenají vysoký gradient, tmavá nebo černá místa značí nízký nebo nulový gradient.



Obrázek 2.4: Gradient běžné fotografie 2.4b, a zdrojová fotografie 2.4a.

Mimochodem je zde také vidět, že gradient funguje také jako detektor hran. Kromě toho ale rovněž zvýraznil šum v obrázku.

2.3.4 Laplaceův operátor

Laplaceův operátor [16] je potřeba zavést, aby bylo možné v kapitole 2.3.5 definovat Poissonovu rovnici. Laplaceův operátor (též nazývaný Laplacián) se označuje symbolem Δ a je definován takto:

$$\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}. \quad (2.6)$$

I když je tuto rovnici samozřejmě možné zobecnit pro n proměnných, rovnice 2.6 je opět ve tvaru pro dvourozměrný eukleidovský prostor, protože tento prostor je pro zpracování

obrazu typický. Uvedené vyjádření se podobá rovnici 2.1 pro definici gradientu. Rozdíl je v tom, že zde jsou parciální derivace „umocněny na druhou“. A skutečně, druhá mocnina operátoru nabla odpovídá Laplaceově operátoru:

$$\Delta = \nabla \cdot \nabla = \nabla^2.$$

Je třeba dát pozor na skutečnost, že výraz Δu se nečte „delta u“, ale „Laplace u“ protože slovo „delta“ je již rezervováno pro přírůstek nějaké veličiny. Tento operátor odpovídá divergenci gradientu. Laplaceův operátor se kromě počítačové grafiky používá také např. ve vlnění, elektrostátice, nebo elektřině a magnetismu. Laplaceův operátor je invariantní vůči posuvům a otáčení.

2.3.5 Poissonova rovnice

Poissonova rovnice [26] je nejjednodušší příklad eliptické parciální rovnice. Jedná se o takovou rovnici, která má na levé straně Laplaceův operátor (2.3.4): $\Delta u = f$, po rozepsání:

$$\Delta u = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} = f(x, y).$$

Existuje také tzv. Laplaceova rovnice, což je speciální případ Poissonovy rovnice:

$$\Delta u = 0.$$

Zatímco Laplaceova rovnice najde uplatnění ve fyzice, např. při výpočtu tepelné výměny těles v prostoru, nehodí se pro použití ve zpracování obrazu. Tam se používá Poissonova rovnice, protože jí lze rekonstruovat obrázek z gradientu.

Řešení Poissonovy rovnice

Ve zpracování obrazu má smysl zabývat se řešením pouze diskrétní Poissonovy rovnice. Jedná se o konečnou diferencii klasické Poissonovy rovnice. Existuje několik metod řešení. Nejpoužívanější jsou [10]: Gaussova eliminační metoda, Jacobiho metoda, metoda sdružených gradientů, *Successive Overrelaxation* (český překlad pravděpodobně neexistuje), rychlá Fourierova transformace a metoda multigrid.

Zde je jejich srovnání z hlediska časové a prostorové složitosti [10]:

Název metody	Časová složitost při sériovém výpočtu	Časová složitost při paralelním výpočtu	Prostorová složitost	Počet procesorů
GEM	$\mathcal{O}(n^2)$	$\mathcal{O}(n)$	$\mathcal{O}(n^{\frac{3}{2}})$	n
JAC	$\mathcal{O}(n^2)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$	n
CG	$\mathcal{O}(n^{\frac{3}{2}})$	$\mathcal{O}(n^{\frac{1}{2}} \cdot \log(n))$	$\mathcal{O}(n)$	n
SOR	$\mathcal{O}(n^{\frac{3}{2}})$	$\mathcal{O}(n^{\frac{1}{2}})$	$\mathcal{O}(n)$	n
FFT	$\mathcal{O}(n \cdot \log(n))$	$\mathcal{O}(\log(n))$	$\mathcal{O}(n)$	n
Multigrid	$\mathcal{O}(n)$	$\mathcal{O}((\log(n))^2)$	$\mathcal{O}(n)$	n

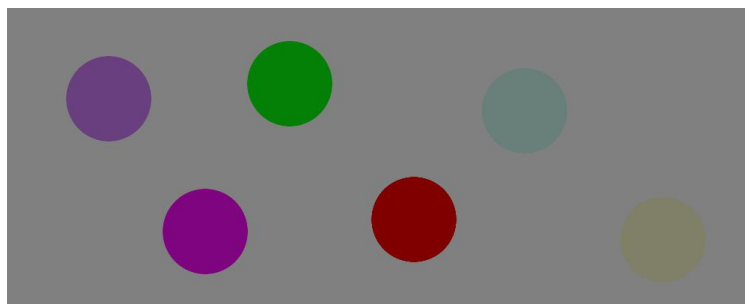
Tabulka 2.2: Srovnání metod pro řešení Poissonovy rovnice.

Význam zkratk: GEM — Gaussova eliminační metoda, JAC — Jacobiho metoda, CG — metoda sdružených gradientů, SOR — Successive Overrelaxation a FFT - metoda založená na rychlé Fourierově transformaci. U hodnot pro paralelní výpočet se předpokládá, že je k dispozici neomezený počet procesorů. Jejich skutečně využitý počet je v posledním sloupci. Z tabulky je vidět, že Gaussova eliminační metoda je velmi pomalá, a proto se v praxi moc nepoužívá. Naopak oblíbené jsou rychlé metody FFT, nebo ještě rychlejší multigríd. Je patrné, že použití paralelní verze algoritmů vede téměř vždy k zrychlení, někdy i o několik tříd.

Nástroje pro řešení Poissonovy rovnice někdy bývají označovány zkratkou PES - Poisson Equation Solver.

2.3.6 Helmholtz-Kohlrausch efekt

Helmholtz-Kohlrausch efekt (často označován zkratkou HK) [33] popisuje vliv odstínu barvy na vnímaný jas. Přestože byl poprvé popsán již v 19. století, nebyl dodnes vědecky vysvětlen. Jeho důsledkem je, že některé barvy (odstíny) lidé vnímají jako tmavší a jiné jako světlejší, přestože ve skutečnosti mají stejnou jasnost. Jedná se tedy o jev založený na lidském vnímání. Přesněji řečeno lidské vnímání světlosti souvisí se sytostí barvy. Většině lidí připadají sytější barvy světlejší, než ty méně syté. To je ilustrováno na obrázku 2.5. První čtyři kruhy zleva jsou poměrně syté a působí proto světleji, než zbývající dva kruhy vpravo, které vypadají, že jsou tmavší. Přesto má celý tento obrázek intenzitu osvětlení 50 %, a to včetně pozadí.



Obrázek 2.5: Isoluminantní obrázek znázorňující subjektivnost lidského vnímání jasu.

Tohoto jevu se využívá při převodu barevných obrázků na černobílé. Při vytváření gradientního pole (2.3.3) je potřeba rozhodnout, kde gradient stoupá a kde klesá, neboli určit jeho znaménko. Možností, jak jej určit, je mnoho a jednou z pokročilých technik je využití právě HK efektu pro výpočet takzvaného HK prediktoru. Opět existuje více možných prediktorů a možností, jak je spočítat. To je popsáno např. v [24]. Yoshinobu Nayatani se v [24] zaměřuje na prediktor jménem Nayatani Variable-Achromatic-Color [24]. Často se používá zkratka Nayatani VAC. Jeho výpočet je formulován takto:

$$\theta = \arctan \frac{v' - v'_c}{u' - u'_c}$$

$$\begin{aligned}
q(\theta) = & -0.01585 \\
& -0.03017\cos(\theta) - 0.04556\cos(2\theta) \\
& -0.02667\cos(3\theta) - 0.00295\cos(4\theta) \\
& +0.14592\sin(\theta) + 0.05084\sin(2\theta) \\
& -0.01900\sin(3\theta) - 0.00764\sin(4\theta)
\end{aligned}$$

$$K_{Br} = 0.2717 \frac{6.469 + 6.362L_a^{0.4495}}{6.469 + L_a^{0.4495}}$$

$$s_{uv}(x, y) = 13((u' - u'_c)^2 + (v' - v'_c)^2)^{\frac{1}{2}}$$

$$\Gamma_{VAC} = 1 + (-0.1340q(\theta) + 0.0872K_{Br})s_{uv}(x, y),$$

kde L_a je adaptivní osvětlení, u' a v' jsou složky u a v barevného modelu CIE Luv (2.1.5), u'_c a v'_c jsou referenční hodnoty bílého bodu. K_{Br} je koeficient určující vliv adaptivního osvětlení (L_a) a $uvv(x, y)$ je metrická saturace testované barvy. Lze vidět, že výpočet není triviální. Takový výpočet nebude nikdy možné s jistotou prohlásit za absolutně správný, právě proto, že je tato technika založena na lidském vnímání a podobných prediktorů tedy existuje více.

2.4 Histogram

Histogram je nástroj pro analýzu obrázků. Histogram udává zastoupení jednotlivých odstínů v obrázku. Někdy je histogram chápán jako sloupcový graf, a velmi často tak bývá zobrazován. To však není přímo podmínkou, histogram lze zobrazit také např. jako spojnicový graf.

Podle tvaru histogramu je pak možné posoudit některé vlastnosti obrázku. Z histogramu lze např. usoudit, že v obrázku některý odstín převládá, nebo že je obrázek vyvážený. Histogram lze zpracovat pro různé kanály obrázku. Nejjednodušší variantou je vytvořit histogram pro obrázek ve stupních šedi. Pokud by zdrojový obrázek byl uložen jako osmi-bitový (každý pixel obrázku by nabýval některého z $2^8 = 256$ možných odstínů), pak by histogram vypadal jako sloupcový graf s 256 sloupci. Na ose X takového histogramu jsou pak jednotlivé odstíny a na ose Y počet těchto odstínů v obrázku. Pro osu Y se u obrázků s velmi silně dominujícím jedním odstínem používá logaritmické měřítko.

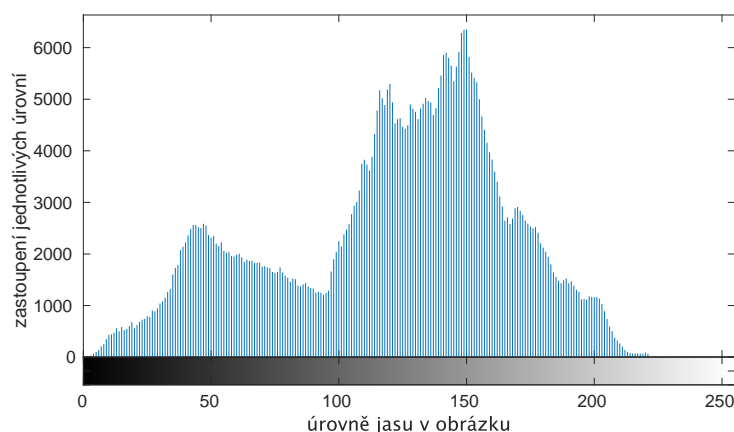
Výpočet histogramu je prostý. V prvním kroku je potřeba vytvořit pole o takové velikosti, kolik různých odstínů je cílené rozlišovat, a toto pole inicializovat na nulu. Pak je třeba projít po bodech vstupní obrázek a inkrementovat jednotlivé indexy v poli podle odstínu právě procházeného bodu. Pokud je cílem porovnávat histogramy dvou obrázků o různých rozměrech, je třeba histogramy normalizovat u obou obrázků na stejnou hodnotu.

Ukázka histogramu je na obrázku 2.7. Na tomto histogramu je specifické, že neobsahuje žádné odstíny s nejvyšším jasem. Naopak je vidět, že vstupní obrázek byl velmi bohatý na střední tóny. Jedná se o histogram obrázku 2.6.

Pomocí histogramu lze např. odhalit vlastnost obrázku zvanou *posterizace*. Tato vlastnost může být někdy záměrná nebo dodatečně vytvořená, většinou se však jedná o vadu obrázku. Posterizace vzniká, např. pokud je obrázek uložen s nedostatečnou bitovou hloubkou. Pak může být na některých barevných přechodech vidět, že přechod z jednoho odstínu do druhého není plynulý, ale zdá se být odstupňovaný, může připomínat vrstevnice. Histogram obrázku postiženého posterizací vypadá potom, jako by byl „děravý“. Odstranění posterizace lze provést speciálními vyhlazovacími filtry.



Obrázek 2.6: Vstupní obrázek pro histogram 2.7.



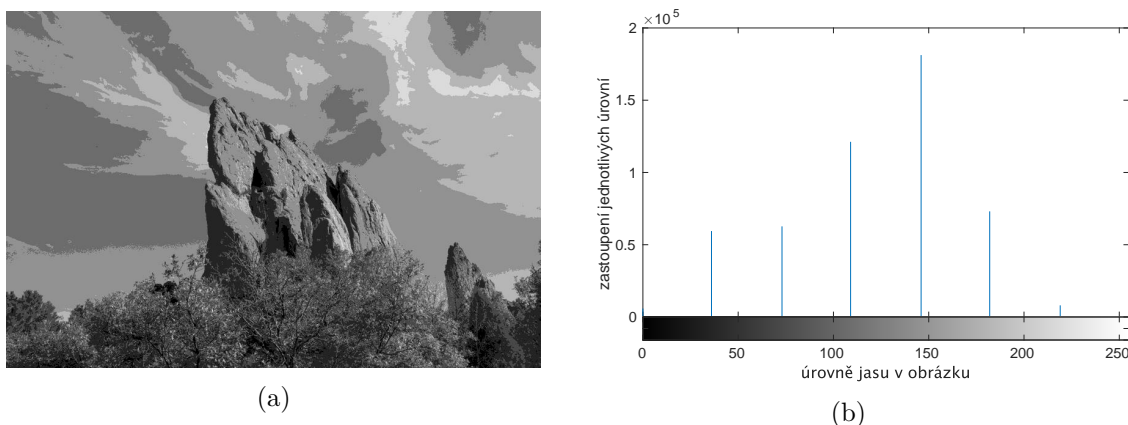
Obrázek 2.7: Ukázka histogramu.

Na obrázku 2.8 je znázorněn poněkud extrémní příklad posterizace. Tento obrázek byl uložen s pouhými třemi bity na pixel a obsahuje tedy pouhých 8 (2^3) odstínů. Jeho histogram je znázorněn pak na obrázku 2.8b. Zde je těchto 8 odstínů pěkně vidět. Sloupců je na obrázku pouze sedm, protože nejjasnější odstín se v obrázku vůbec nevyskytuje.

2.5 Histogram matching

Histogram matching je metoda přenosu vzhledu z jednoho obrázku na druhý. Jak název napovídá, jedná se o přenos histogramu. Tím dochází nejčastěji k přenosu zastoupení jednotlivých odstínů v obrázku.

Postup aplikace histogram matching je poměrně jednoduchý. Nejprve je třeba vytvořit histogramy vstupního a modelového obrázku. Z těchto histogramů je poté nutné vypočítat distribuční funkce (kumulovaný histogram/kumulovanou pravděpodobnost). Tyto distribuční funkce je třeba normalizovat na stejnou hodnotu, jinak by se operace neprovedla správně. Poté je možné začít se samotným přenosem. Pro každý vstupní bod se použije



Obrázek 2.8: Ukázka obrázku postiženého posterizací (2.8a) a jeho histogram (2.8b).

jeho intenzita a k té se nalezne hodnota na jeho distribuční funkci. Tato nalezená hodnota je pak hledána na distribuční funkci modelového obrázku a nalezený index tvoří výslednou intenzitu. V reálných případech si hodnoty na distribučních funkcích přesně neodpovídají a je proto zapotřebí nalézt nejbližší hodnotu.

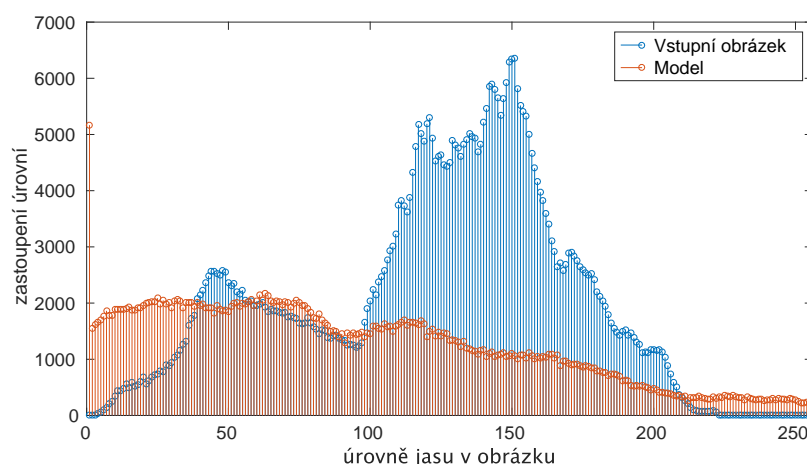
Následovat budou obrázky s příkladem této operace. Obrázek 2.9 znázorňuje vstupní fotografii spolu s modelem. Na následujícím obrázku 2.10 jsou pak přes sebe histogramy těchto dvou fotografií. Z histogramů je patrné, že zatímco vstupní obrázek je bohatý na střední tóny a postrádá světlé odstíny, modelový obrázek je poměrně vyvážený s dominancí tmavších tónů. Obrázek 2.11 zobrazuje srovnání distribučních funkcí těchto histogramů. Na obrázku 2.12 je pak výsledek po provedení histogram matching.



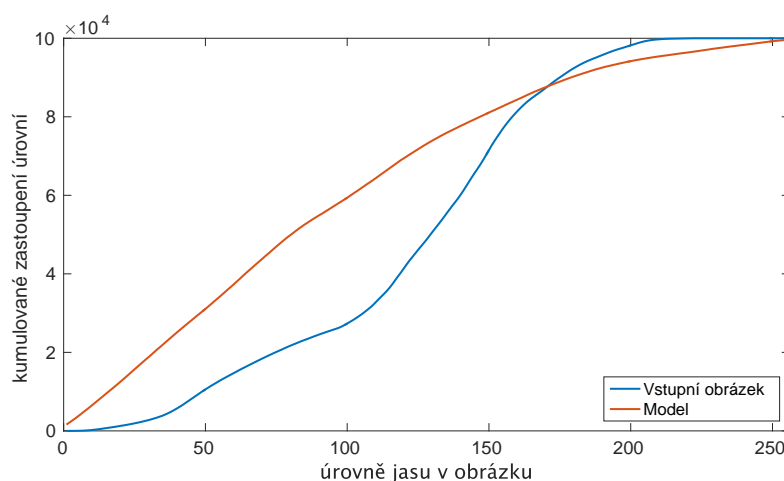
Obrázek 2.9: Vstupní obrázek (2.9a) a modelový obrázek (2.9b) pro provedení histogram matching.

2.6 HDR

HDR [7, 14, 6] je zkratka pro High Dynamic Range, což v překladu znamená vysoký dynamický rozsah. Někdy se používá též varianta HDRI tj. High Dynamic Range Imaging.



Obrázek 2.10: Srovnání histogramů vstupního obrázku a modelového obrázku.



Obrázek 2.11: Srovnání distribučních funkcí vstupního a modelového obrázku.

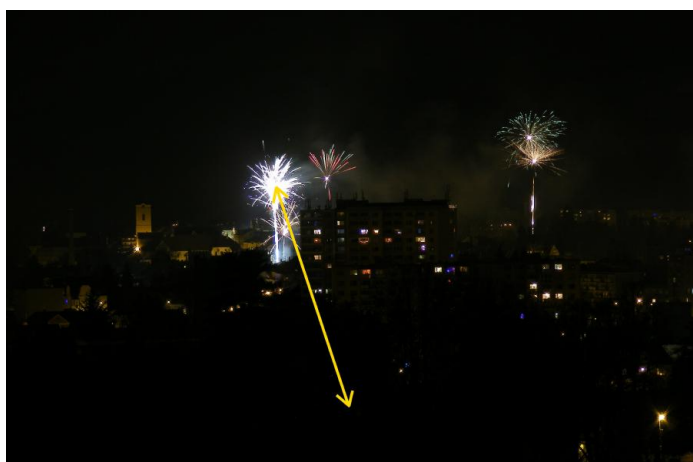
Dynamický rozsah je obecně podíl mezi nejvyšší a nejnižší hodnotou na určité stupnici. Konkrétně v digitální fotografii se jedná o poměr mezi nejjasnějším a nejtemnějším bodem v obrázku. To je znázorněno na obrázku 2.13 pomocí žluté šipky.

Množství světla je nutné nějakým způsobem měřit. Jednou ze sedmi základních jednotek soustavy SI je Kandela [28], značka cd. Kandela je jednotkou svítivosti, což je fotometrická veličina s označením I. Jednotka cd odpovídá přibližně svítivosti jedné klasické svíčky. Běžná wolframová 100 W žárovka má svítivost přibližně 150 cd, reflektor v autě pak až 100 000 cd. Příbuznou veličinou je světelný tok, někdy též označovaný jako světelný výkon, který je udáván v lumenech. Převod mezi těmito dvěma veličinami je nutné znát úhel rotačního kužele zdroje světla. Další fotometrickou veličinou je jas, který se značí písmenem L. Jeho vyjádření se opírá o svítivost, ke které však přidává ještě závislost na ploše povrchu světelného zdroje. Jednotkou jasu je tedy cd/m^2 . Je dobré vědět, že cd/m^2 přesně odpovídá jednotce nit, protože někteří výrobci displejů udávají jas právě v nitech.

Přibližné hodnoty jasu pro vybrané světelné zdroje jsou zapsány v tabulce 2.3



Obrázek 2.12: Výstupní obrázek.



Obrázek 2.13: Presentace pojmu dynamický rozsah. Žlutá šipka spojuje nejsvětlejší a nejtmavší bod obrázku.

Z uvedené tabulky by mělo být patrné, že rozsah jasu není přímo závislý na konkrétních zdrojích světla. Na příklad rozsah jasu 1000:1 může nastat mezi měsíčním světlem a místností osvětlenou denním světlem, ale stejně tak může nastat mezi jasným slunečním světlem a slunečním diskem. Právě to je důvod, proč se o rozsahu mluví jako o dynamickém. Dynamický rozsah se označuje slovem vysoký, protože v extrémních situacích se lze v běžných podmínkách na Zemi setkat s dynamickým rozsahem až $10^{12} : 1$.

2.6.1 Pořizování HDR snímků

Předchozí odstavec popisoval jas různých objektů z reálného světa. Lidské oči jsou extrémně citlivé na světlo a jejich pracovní rozsah je díky adaptačním mechanismům obrovský. Člověk se zdravým zrakem dokáže přečíst vytištěný text v místnosti osvětlené pouze monitorem a zároveň bez problému rozlišit všechny detaily na tomto monitoru. Jeden z velkých problémů současné fotografie je, že dnešní digitální snímače mají rozsah expozice o několik

Zdroj světla	Jas [cd/m^2]
Světlo hvězd	10^{-3}
Měsíční světlo	10^{-1}
Osvětlení v místnosti	10^2
Zataženo	$2 \cdot 10^3$
Jasno	10^4
Sluneční disk	10^8

Tabulka 2.3: Hodnoty jasu pro jednotlivé světelné zdroje.

řádů nižší než lidské oko. V tomto ohledu byla velmi podobná situace i v době používání filmů. Rozsah expozice je omezen kapacitou CCD buněk na snímacím čipu.

Klasické kompaktní fotoaparáty mají dynamický rozsah přibližně 4-5 EV, u klasických filmů to bylo zhruba 6-7 EV, u profesionálních fotoaparátů je tato hodnota až 14 EV. EV znamená Exposure Value, neboli expoziční jednotka. Tato jednotka jaksi zapouzdřuje všechny důležité fotografické parametry snímku do jednoho čísla. Důležité je, že EV udává množství světla, na které je fotoaparát při fotografování nastaven. Stupnice EV je logaritmická. To znamená, že např. +1 EV nebo +2 EV odpovídá oproti 0 EV dvojnásobnému, respektive čtyřnásobnému množství světla. Z těchto hodnot je možné udělat si představu o tom, jak nízký je dynamický rozsah dnešních snímačů, zvláště ve srovnání s přibližným maximálním dynamickým rozsahem v přírodě $10^{12} : 1$.

Důsledkem nízkého dynamického rozsahu je, že při pokusu o vytvoření fotografie scény s vysokým dynamickým rozsahem dnešními fotoaparáty je scéna buď přexponovaná — obsahuje bílé plochy bez detailů, nebo při jiném nastavení fotoaparátu je podexponovaná — obsahuje černé plochy bez detailů. Jinými slovy taková fotografie neobsahuje detaily v celém rozsahu. Právě tuto situaci znázorňuje následující obrázek 2.14.



Obrázek 2.14: Fotografie, které vzniknou při fotografování scény s vysokým dynamickým rozsahem fotoaparátem s nízkým dynamickým rozsahem. Výsledkem je fotografie buď přexponovaná (2.14a) nebo podexponovaná (2.14b).

Fotograf ovlivňuje expozici nastavením parametrů fotoaparátu. Na obrázku 2.14a preferoval fotograf prokreslené stíny, které obsahují mnoho detailů. Daní za toto nastavení je

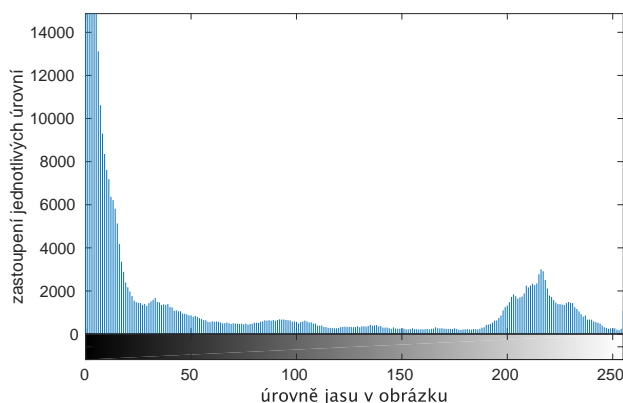
ale obloha, která se jeví docela přepálená a obsahuje jen málo detailů. Druhý obrázek 2.14b představuje opačnou situaci. Fotoaparát byl nastaven tak, aby detailně prokreslil oblohu. Tyto parametry však způsobily tmavé, až téměř černé oblasti v dolní části obrázku. Cílem může být z různých důvodů také např. zachytit mračna výraznější než ve skutečnosti jsou. Podstatné je, že neexistuje nastavení fotoaparátu s tímto dynamickým rozsahem, které by tuto scénu zachytilo rovnoměrně a detailně ve všech oblastech.

Velmi podobné problémy se řešily nebo stále řeší fotografové i při fotografování na film. U klasické analogové fotografie ale nejsou možnosti řešení příliš bohaté. Jednou z možností by mohlo být použití přechodového filtru. Přechodový filtr je fyzická pomůcka, která se upevní na fotoaparát a ovlivňuje vzhled výsledných fotografií. Přechodový filtr je tvořen sklem, které má dvě poloviny s různou propustností světla. Jedna polovina je typicky viditelně zatmavená, zatímco druhá nebývá zatmavená vůbec. Přechod mezi těmito dvěma segmenty je samozřejmě plynulý, aby na výsledné fotografii nebyl pokud možno zpozorovatelný. Tento filtr by při vytváření fotografie 2.14a skutečně mohl přinést požadovaný výsledek. Nevýhodou přechodových filtrů však je, že je lze použít pro specifický typ scény, typicky pro exteriér s viditelnou jasnou oblohou. Významná je skutečnost, že snímaná krajina by měla mít pokud možno minimálně členitý horizont, jinak může být přechod viditelný. Přechodové filtry se často používají při fotografování na film, ale nic nebrání jejich použití i s digitálními fotoaparáty.

Druhou možností, jak lze nedostatečný dynamický rozsah snímačů nebo filmu vyřešit, je použití blesku nebo jiného světelného zdroje. To je patrné na fotografii 2.15, která je vyobrazena i se svým histogramem. Pozadí obrázku je dobře prokreslené, zatímco osoba v popředí je podexponovaná a nelze téměř rozeznat. Při použití blesku by došlo k prosvětlení popředí při zachování detailů v pozadí.



(a)



(b)

Obrázek 2.15: Fotografie, u které by nedostatečný dynamický rozsah snímače mohlo vyřešit použití blesku 2.15a. Vpravo (2.15b) je pak histogram této fotografie

Blesk lze však použít pouze v omezeném rozsahu případů. Kromě typických omezení s dosahem může také blesk např. nežádoucím způsobem odrážet vlhkost ve vzduchu atd.

Současná digitální technika nabízí velmi inovativní přístup k řešení tohoto problému a tím je právě tvorba HDR snímků. Princip je poměrně jednoduchý. Spočívá ve vytvoření

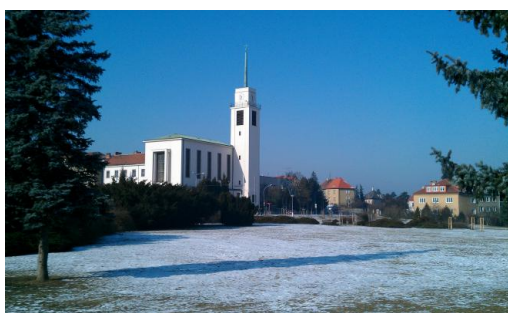
série snímků se stejnou polohou fotoaparátu, avšak s různým rozdělením expozice, a následném spojení této série do jednoho obrázku. Pro vytváření různě exponovaných fotografií by teoreticky bylo možné upravovat kterýkoliv ze tří základních parametrů fotoaparátu: expoziční čas, clonové číslo a citlivost na světlo (ISO). Při změně clonového čísla však dochází ke změně hloubky ostrosti a to by mohlo znehodnotit pokusy o sloučení série fotografií. Při změně ISO (při zvyšování) zase dochází k nárůstu šumu v obraze, což je téměř vždy nežádoucí. Parametr, který je tedy vhodný pro upravování, je expoziční čas. Velmi vhodné je při tomto procesu použít stativ. Výhodné je též použít dálkovou spoušť, protože při změně parametrů přímo na těle fotoaparátu může dojít k nechtěnému pohybu s fotoaparátem. Takto způsobený posun na fotografii je sice softwarově řešitelný, vždy to ale vede k snížení kvality.

Kromě popsané techniky existují také speciální HDR kamery, např. SpheronVR.

2.6.2 Kdy je HDR potřeba

Z předchozích odstavců by mělo být zřetelné, že HDR není třeba používat vždy. Při fotografování některých scén by HDR nepřineslo žádný užitek. Typicky problémové scény, kde je HDR výhodné, jsou např. krajina anebo fotografování, při kterém byl fotoaparát namířen směrem ke slunci. Dalším příkladem může být scéna v interiéru, na které je v záběru okno. Do relativně tmavého interiéru pak proniká oknem velké množství světla a tím vzniká vysoký dynamický rozsah jasů. Dalším kandidátem pro použití HDR jsou noční scény, ve kterých se vyskytuje nějaký silný zdroj světla, např. pouliční lampa. Aby noční scéna nebyla podexponovaná, je většinou potřeba nastavit dlouhý expoziční čas. To ale způsobí nepřírozené zvýraznění zdroje světla a HDR se tedy přímo nabízí. Obecně se jedná o všechny scény obsahující protisvětlo.

Existují také scény, kde by HDR nemělo žádný přínos. Takovými jsou např. scéna v exteriéru, ale bez viditelné oblohy, nebo scéna v exteriéru se Sluncem za zády fotografa. Dále sem patří např. zátiší nebo makro fotografie. Ukázka těchto dvou kategorií je na obrázku 2.16



(a)



(b)

Obrázek 2.16: Ukázka obrázku, kde HDR není zapotřebí 2.16a a obrázku, kde najde HDR uplatnění 2.16b.

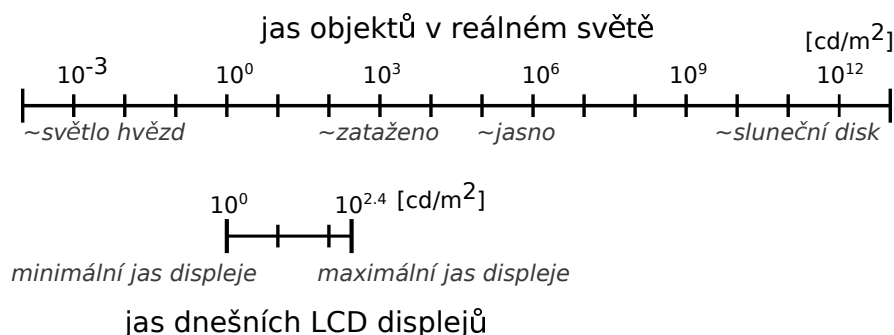
Poslední kategorií jsou takové scény, pro které je HDR (vytvořené z několika expozic) vyloženě nevhodné a může kvalitu fotografie zhoršit. Sem patří fotografie pohybujících se objektů. Zde HDR prakticky použít nelze, protože pokud dojde mezi vytvořením dvou fotografií k pohybu fotografovaného objektu, vznikne při složení na výsledné fotografii ne-

žádoucí efekt, tzv. „duch“. V některých situacích je tento problém obtížně řešitelný, např. při pohybu oblaků. Existují sice způsoby jak se s tímto vyrovnat (anti-ghosting), obecně je to však na škodu.

Údaj o tom, zda je vhodné HDR použít, lze snadno vyčíst z histogramu. To je vidět na obrázku 2.15. Histogram ukazuje, že většina bodů je koncentrována do krajních částí histogramu, což je známkou potřeby použití HDR. Je zde ještě jedna možnost, jak lze nutnost použití HDR zmírnit. Touto volbou je ukládání do formátu RAW. Tento formát pracuje oproti běžným, silně komprimovaným formátům, s větší bitovou hloubkou a tím pozitivně ovlivňuje dynamický rozsah. Samozřejmě je ale celé toto snažení závislé na použitém snímacím čipu. Pokud snímač sám o sobě nezvládá vysoký dynamický rozsah, pak tuto situaci typicky nelze formátem RAW zachránit.

2.6.3 Zobrazování HDR obsahu

Pomocí postupu popsaného v 2.6.1 lze tedy v paměti počítače s dobrou úspěšností vytvořit HDR obrázek. Informaci o jasů bodu je možné ukládat např. v pohyblivé desetinné čárce s dvojitou přesností. To je dostatečný rozsah pro velmi kvalitní HDR. Další problém ale nastává při pokusu o zobrazení HDR obrázku. Současné běžně prodávané displeje jsou totiž LDR (LDR je zkratka pro low dynamic range, tedy označení pro „obyčejný“ dynamický rozsah). Přesněji řečeno, dnešní LCD displeje dokáží produkovat světlo se svítivostí přibližně od 1 do 250 cd/m^2 . Srovnání této svítivosti a svítivosti objektů v reálném světě je na obrázku 2.17.



Obrázek 2.17: Porovnání svítivosti dnešních displejů a skutečných objektů.

Přirozeně ještě o něco horší situace je při pokusu o tisk HDR obrázku. Papír pouze odráží dopadající světlo a nemá tedy žádnou svítivost. Lze u něj ale určit poměr jasů. U obyčejných kancelářských papírů je to přibližně 50:1, u fotopapíru pak kolem 100:1 [14].

Nejjednodušší možností, jak zobrazit HDR obrázek na LDR displeji je stupnici uvedenu na obrázku 2.17 prostě přemapovat. To znamená nejprve rozsah posunout o nějakou konstantu (přibližně by to bylo o 10^3) a poté vydělit nějakou konstantou. Takovou konstantou by mohlo být při velmi hrubém výpočtu číslo kolem 10^{12} . Tento postup funguje a v zobrazeném obrázku jsou skutečně vidět detaily jak v jasných, tak v tmavých oblastech. Problematické ale je významné snížení kontrastu obrázku. Takto zobrazený obrázek pak působí mdle, vybledle. Tato metoda se nezývá lineární mapování.

Proces, při kterém dochází k převodu obrázků s vysokým dynamickým rozsahem do běžně zobrazitelné formy, se nazývá tone mapping. Jednotlivé metody pro provedení tone mappingu se pak označují jako operátory.

Vývoj se však posouvá vpřed i v této oblasti, v oblasti vývoje HDR displejů. Na trhu se již začínají objevovat první HDR displeje a objevují se také zprávy o HDR displeji v telefonu (stav březen 2016). HDR displeje se od těch běžných odlišují jednou zásadní věcí, a tou je, že podporují vyšší jas. Zmíněný mobilní telefon má mít displej s maximálním jasnem až 1300 cd/m^2 (nitů). V praxi by to mělo znamenat, že pokud se na displeji objeví např. záběr na slunce nebo rozpálenou žárovku, pozorovatel bude muset přimhouřit oči. To odpovídá realitě, protože reálně není možné tyto objekty nechráněnými očima pozorovat. Tato technologie však ještě bude muset projít vývojem. Představené HDR televizory mají spotřebu v řádu stovek Wattů a je nutné je chladit ventilátory. Většímu rozšíření také brání vysoká cena. Na výsledný dojem bude mít velký vliv také technologie podsvícení panelu. Nejjednodušší a nejpoužívanější varianta se nazývá „edge LED“. U této technologie jsou světelné diody umístěny po jedné nebo více hranách zobrazovacího panelu a světlo z nich je na plochu panelu rozváděno pomocí speciálních světlovodů. Systém je vyroben tak, že je vždy podsvícena celá plocha panelu. To je největší nevýhoda tohoto řešení, protože to znamená, že i body, které by měly být úplně černé, jsou podsvícené. S touto technologií tedy není možné zobrazit příliš hlubokou černou. Podstatně lépe je na tom technologie „direct LED“. Zde jsou světlo emitující diody umístěny přímo za LCD panelem. Těchto podsvětlovacích diod se používá řádově méně, než jaké je rozlišení panelu. Podsvícení je tedy rozděleno do jakýchsi segmentů. Je zde ale možné podsvícení lokálně regulovat a tím dosáhnout lepšího podání černé barvy. Dosud nejdokonalejší technologií je technologie OLED. Ta se od předchozích dvou výrazně liší, protože edge i direct LED jsou pouze systémy podsvícení LCD panelu, zatímco OLED je technologie samotného displeje. Největší výhodou OLEDu je, že světlo produkují samotné zobrazovací buňky a není zde tedy potřeba žádné podsvětlení. Z toho vyplývá možnost zobrazit opravdu velmi tmavou černou, a tím pádem poskytnout velmi vysoký kontrast. HDR displeje budou mít jistě nejlepší parametry ve spojení s technologií OLED, zatím jsou však tyto displeje neúměrně drahé.

Používáním techniky HDR je známý např. americký fotograf Tom Cooper [9].

2.7 Gaussův filtr

Šum [34] je obecně nežádoucí znečištění určitého signálu. Je to nejčastěji nežádoucí produkt vznikající při přenosu, nebo už při samotném získávání, anebo kódování informací. Někdy však může být šum i záměrně generován, např. v oboru procedurálního generování. Na konci minulého století bylo možné se často setkat s šumem v analogových televizorech, kde šum vznikal přenosem za použití inverzní amplitudové modulace. Dnes se s šumem setkáme nejčastěji pravděpodobně při prohlížení digitálních fotografií, kde je šum velmi častý zvláště při použití vysokého ISO. Tento typ šumu se někdy označuje jako náhodný šum.

K odstranění šumu je nutné znát jeho charakteristiku. Pro odšumování obrázků se používá velké množství filtrů. Tyto filtry lze rozdělit na lineární a nelineární. Aplikují se na zašuměný obrázek např. pomocí konvoluce. Filtr reprezentovaný konvoluční maticí lze tedy chápat jako funkci dvou proměnných.

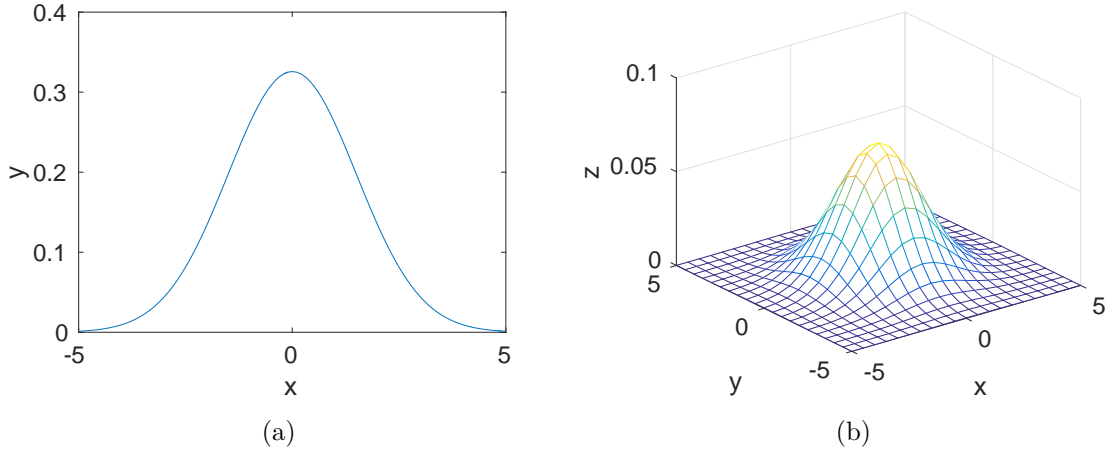
Jedním z nejjednodušších filtrů je Gaussův filtr. Někdy je též nazýván Gaussovo rozmazání, protože to je právě to, co tento filtr dělá. Gaussova funkce je definována takto[12]:

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \quad (2.7)$$

Ve zpracování obrázku však najde uplatnění spíše 2D varianta této funkce, která vypadá takto:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \quad (2.8)$$

kde σ je parametr nazývaný směrodatná odchylka. „Šířka grafu“ je právě na této odchylce závislá, je jí přímo úměrná. Uvedené předpisy (2.7, 2.8) jsou již v normalizovaném tvaru a díky tomu je jejich plocha (integrál) rovna jedné. Grafy těchto funkcí jsou zobrazeny na obrázku 2.18.



Obrázek 2.18: Grafy Gaussovy funkce. Varianta 1D (2.18a) a 2D (2.18b), $\sigma = 1.5$ v obou případech.

Důležité je používat rovnice v normalizovaném tvaru (což 2.8 je), protože jinak by docházelo ke vzniku artefaktů. Princip činnosti Gaussova filtru je velmi jednoduchý. Hodnota každého pixelu odpovídá váženému průměru okolních pixelů, přičemž váha každého pixelu je dána právě Gaussovou funkcí. Přirozeně, pixel ve středu filtračního okna má nejvyšší váhu a směrem ke krajům se jeho vliv snižuje. Důsledkem takového filtrování tedy skutečně je, že drobný šum je odfiltrován. Spolu s ním jsou však odfiltrovány také všechny ostatní detaily a tím dochází k rozmazání obrázku. Zvláště důležité je proto správně nastavit parametr σ , který přímo ovlivňuje sílu filtru. Při příliš nízkém σ bude efekt filtru nízký a bude tedy částečně zachován i šum, a naopak při příliš vysokém σ obrázek ztratí hodně ze své ostrosti.

2.8 Bilaterální filtr

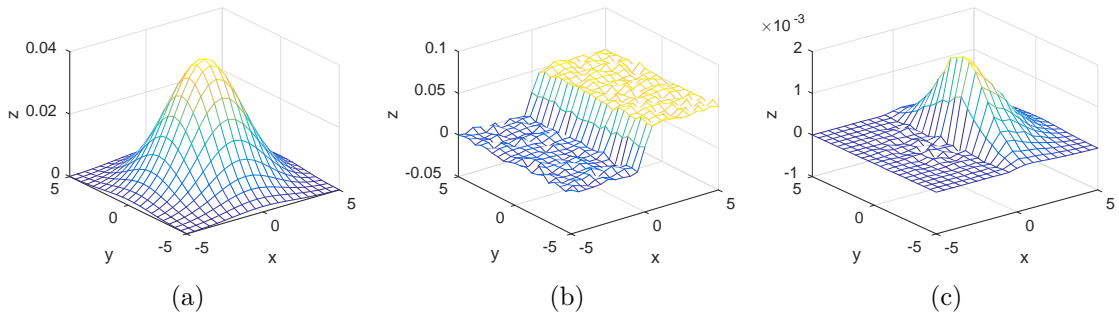
Rozšířením Gaussova filtru vznikl bilaterální filtr [25, 30]. Jeho největší předností je, že zachovává ostré hrany. Toho bylo dosaženo tím, že tento filtr neprůměruje přes hrany. To je důsledek toho, že tvar jádra závisí na obsahu obrázku. Gaussov filtr (2.7) má pouze jediný parametr. Je jím směrodatná odchylka Gaussovy funkce, a funguje jako pro nastavení „prostorové váhy“, se kterou jádro pracuje. Bilaterální filtr k tomuto parametru přidává ještě druhý parametr pro určení váhy rozsahu, s jakým se právě filtruje. Diskrétní konvoluce filtru je definována takto:

$$bf(I)_p = \frac{1}{W_p} \sum_{q \in S} g_{\sigma_s}(\|p - q\|) g_{\sigma_r}(\|I_p - I_q\|) I_q, \quad (2.9)$$

kde:

$$W_p = \sum_{q \in I} g_{\sigma_s}(\|p - q\|) g_{\sigma_r}(|I_p - I_q|).$$

Důležité je zmínit, že $g_{\sigma}(x)$ je Gaussova funkce ve tvaru $g_{\sigma}(x) = e^{-\frac{x^2}{\sigma^2}}$. W_p je normalizační část. σ_s je parametr ovlivňující prostorovou váhu, a odpovídá parametru σ u Gaussova filtru. σ_r je nový parametr ovlivňující váhu rozsahu. I je vstupní obrázek a p je aktuálně zkoumaný pixel. Význam parametru σ_s již byl vysvětlen v 2.7, pro bilaterální filtr je však důležitější parametr σ_r . Ten také ovlivňuje nastavení Gaussovy funkce, ale tentokrát je to funkce, kterou se hodnotí váha daného odstínu. Nejen tedy, že se pro každý pixel počítá jeho váha v závislosti na poloze ve filtrovacím okně, počítá se pro něj také váha v závislosti na jeho odstínu. Pokud je parametr σ_r příliš vysoký, pak tento filtr degraduje do funkce Gaussova filtru, protože všechny intenzity z rozsahu budou mít stejnou váhu. S tím, jak se parametr σ_r snižuje, snižuje se také „šířka“ rozsahu s významnou vahou. To jinými slovy znamená, že pixely s výrazně odlišným odstínem než zkoumaný pixel mají nižší váhu. To je přesně princip toho, jak tento filtr zachovává hrany. Pokud je σ_r příliš nízké, hrozí že filtr nebude účinný.



Obrázek 2.19: Tvorba jádra (kernelu) bilaterálního filtru. Prostorová část 2.19a, rozsahová část 2.19b, a tyto dvě části zkombinované do výsledného filtrovacího jádra 2.19c.

Obrázek 2.19 ukazuje tvorbu filtrovacího jádra a tím také funkci bilaterálního filtru. Na obrázku 2.19a je vidět, že parametr σ_s byl nastaven na poměrně vysoké číslo, a proto je základní graf široký – vliv okolních pixelů je významný. Z obrázku 2.19b je pak vidět, že filtrovací jádro se právě nacházelo na hraně v obrázku a zároveň byl parametr σ_r nastaven na nízké číslo – toto vedlo k vytvoření onoho „schodu“. Na obrázku 2.19c, který zobrazuje výsledné filtrovací jádro, jsou pak tyto dvě části zkombinované dohromady. Je důležité si uvědomit, že takto jádro vypadá právě na jedné pozici v obrázku, a že s posunem na jiné souřadnice se toto jádro mění.

Obrázek 2.22 názorně ukazuje vliv parametrů σ_s a σ_r na výsledek filtrování.

Rozměry obrázku zdrojového obrázku k sérii 2.22 byly záměrně sníženy na hodnotu 500x271 bodů, aby více vynikly detaily. Z obrázku 2.22 je také patrné, že např. pro parametry $\sigma_s = 2$, $\sigma_r = 50$ byla velká část detailů odstraněna, ale přitom byly zachovány významné hrany (např. na hlavě ptáka, nebo na trávě v popředí). Jemnějším nastavením parametrů by bylo možné dosáhnout ještě lepších výsledků.

2.9 Chyby obrazu

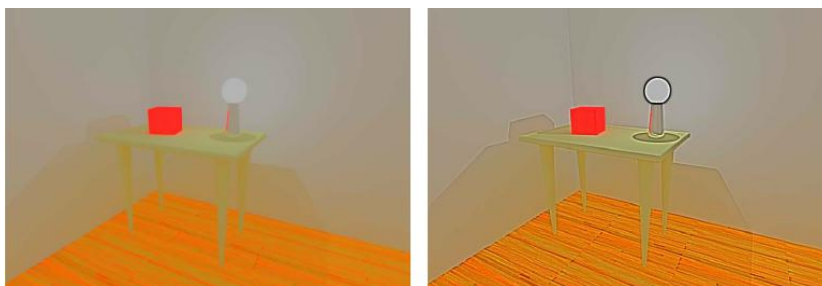
Tato stručná kapitola obsahuje popis některých chyb obrazu. Zaměřuje se na chyby, které mohou vznikat transformacemi obrázků.

2.9.1 Svatozář

Svatozář (halo efekt) [11] je artefakt vznikající například v důsledku použití dvou obrazů filtrovaných dvěma různě velikými filtry. Bývá popisován jako „aura“ kolem hrany v obrázku.

Při zostřování obrázku může někdy vznikat jistá forma halo efektu blízko hran přímo v rámci procesu zostřování. Částečně lze vznik halo efektu omezit použitím nelineárních vyhlazovacích filtrů zachovávajících hrany, jako např. anisotropic diffusion. Jednou z možností vyhnutí se vzniku halo efektu je také práce v gradientní doméně.

Ukázka halo efektu je na obrázku 2.20.



Obrázek 2.20: Ukázka dvou variant halo efektu³.

2.9.2 Chyby způsobené kompresí JPEG








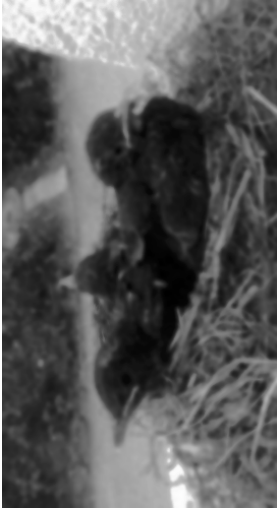

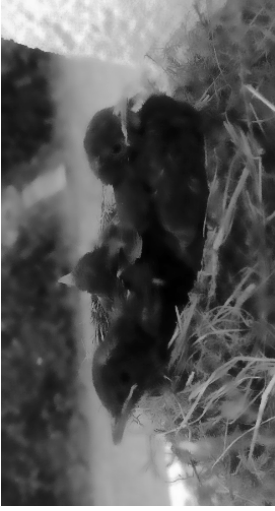

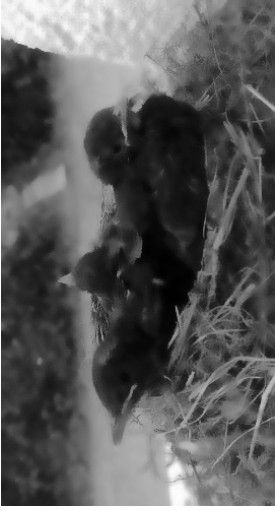
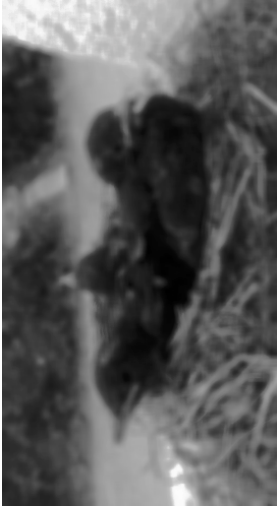
Kompresní metoda JPEG je ztrátová s nastavitelným stupněm kvality. Vstupní obrázek je rozdělen do bloků 8×8 , na které je poté aplikována dvourozměrná diskretní kosinová transformace. Pokud je však kvalita nastavena na příliš nízkou (cílem je vysoký kompresní poměr), mohou být hranice mezi jednotlivými bloky viditelné. Právě to ukazuje obrázek 2.21



Obrázek 2.21: Artefakty v obraze způsobené příliš silnou JPEG kompresí.

³Obrázek byl převzat z

http://www.researchgate.net/profile/Alessandro_Artusi/publication/223680281/figure/fig4/AS:305062451662851@1449743964479/Fig-4-Halo-artifacts-and-oversaturation-Left-HDR-image-after-successful-tone-mapping.png

$\sigma_s = 1$			$\sigma_s = 2$	$\sigma_s = 5$	
					$\sigma_r = 5$
					$\sigma_r = 50$
					$\sigma_r = 255$

Obrázek 2.22: Ukázka vlivu parametrů σ_s a σ_r na přefiltrovaný obrázek.

Kapitola 3

Návrh a implementace

Účelem této kapitoly je popsat principy metod zvolených pro implementaci, navržené řešení a samotný kód. Ústřední částí této kapitoly je část Implementované metody 3.1. Ke každé metodě je zde nejprve teoretický popis její činnosti, a dále následují poznámky k implementaci. Celý projekt byl zakomponován do již existujícího systému s názvem „Tone mapping studio“, který je popsán v kapitole 3.2. Jako první byla implementována podpora barevných modelů a hlavně převodů mezi nimi, tomuto se věnuje kapitola 3.3. Následující kapitola 3.4 obsahuje velmi stručné seznámení s použitými knihovnami. Na závěr této kapitoly je zmíněno použité vybavení 3.5, to se částečně vztahuje také ke kapitole 4.

Vývoj probíhal za pomoci verzovacího nástroje Git. Konkrétně byla použita webová služba GitHub¹. Odkaz na repozitář se zdrojovými soubory se nachází v poznámce². Kromě toho jsou samozřejmě všechny zdrojové soubory umístěny na přiloženém DVD (A).

3.1 Implementované metody

Následovat bude popis implementovaných metod. Každá z následujících tří kapitol obsahuje nejprve vysvětlení principu metody, a poté samotný popis implementace.

Metody pro převod barevných obrázků na černobílé lze rozdělit na lokální a globální. Základní rozdíl je v tom, že u globálních metod existuje převodní křivka, která je stejná pro celý obrázek, což neplatí pro lokální metody. U lokálních metod bývá převodní křivka ovlivněna např. okolím bodu. Z toho také vyplývají výhody a nevýhody obou řešení. Hlavní nevýhodou lokálních metod je, že dva body se stejnou barvou na vstupu mohou vést na dva body s různou barvou na výstupu. To se samozřejmě nemůže stát u globálních metod, protože tam se zpracovává celý obrázek podle stejných pravidel. Dalšími výhodami globálních metod je, že bývají většinou rychlejší a také působí přirozeně. Nevýhodou lokálních metod je to, že mohou produkovat halo artefakty (2.9.1). Výhodou lokálních metod pak je, že mají větší možnosti při převodu – jsou schopné reprodukovat originál přesněji.

Pro zjednodušení orientace jak v zdrojových textech, tak pro kratší odkazování v tomto dokumentu byly v této práci zavedeny zkrácené názvy podle příjmení jednoho z autorů a roku vytvoření metody. Tyto zkrácené názvy jsou: *Kim09* pro metodu *Robust Color-to-gray via Nonlinear Global Mapping* (3.1.1), *Zheng15* pro metodu *Efficient Color-to-Gray Conversion for Digital Images in Gradient Domain* (3.1.2) a *Bae06* pro metodu *Two-scale Tone Management for Photographic Look* (3.1.3).

¹[//github.com/](https://github.com/)

²[//github.com/cadik/TMS](https://github.com/cadik/TMS)

Zatímco metoda Kim09 (3.1.1) je globální, metody Zheng15(3.1.2) a Bae06 (3.1.3) jsou lokální.

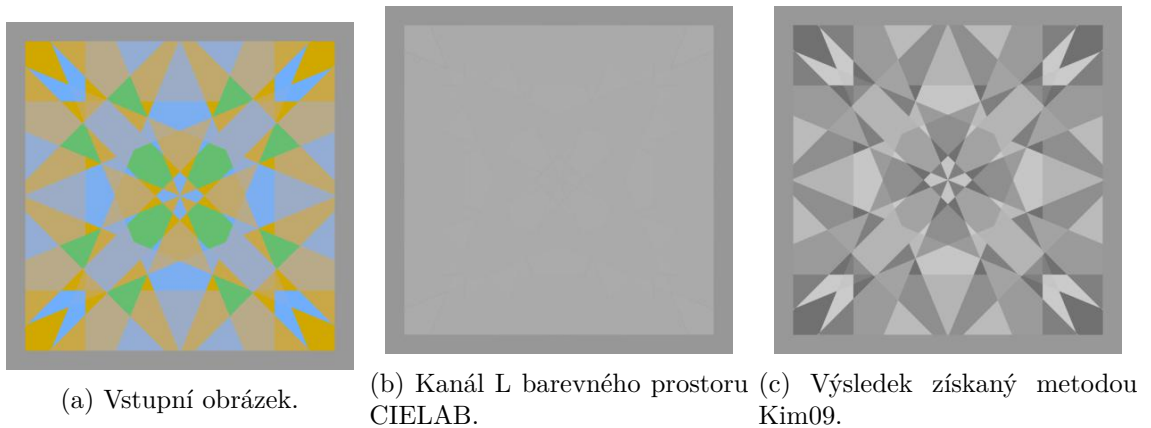
Při implementaci nebyl kladen nijak zvýšený důraz na efektivitu implementace. Cílem této práce je srovnat výsledky implementovaných metod a na ně případná rychlost algoritmu nemá vliv. Všechny metody ale byly implementovány s přibližně stejným stupněm optimalizace, a proto je jejich vzájemné srovnání z hlediska rychlosti možné. Optimalizovat by bylo možné např. předpočítáním určitých hodnot, nebo použitím rychlejšího přístupu do pole, to zde však nebylo prioritou.

3.1.1 Robust Color-to-gray via Nonlinear Global Mapping

Metoda Robust Color-to-gray via Nonlinear Global Mapping [18, 33] je rychlá globální metoda pro převod barevných obrázků na černobílé.

Jednou z jejích vlastností je, že zachovává rozlišitelnost významných rysů obrázku. Dále se snaží zachovávat světlost jednotlivých bodů. Tato metody si také bere za cíl být rychlá a robustní. Cílí na to, aby převedené obrázky měly vizuální vzhled jako barevné obrázky. Rozšíření této metody, také popsané v [18], umí převádět dočasně souvislé video. Této části práce se však tento projekt nevěnuje.

Díky tomu, že je tato metoda nelineární, dovede si poradit s isoluminantními obrázky, jak je vidět na obrázku 3.1. Nelineárnost je zde pojata formou převodové funkce, která definuje, na jaký odstín se převede jaká barva.



Obrázek 3.1: Srovnání černobílých obrázků získaných z kanálu L 3.1b a z metody Kim09 3.1c ze zdrojového obrázku 3.1a

Princip činnosti

Základním barevným modelem pro tuto metodu je model LCH. Dále však tato metoda využívá i modely CIE $L^*a^*b^*$ a CIE Luv. Jedním z cílů metody je dodržení konzistence mapování. To znamená, že stejné barvy na vstupu budou převedeny na stejný odstín na výstupu. Toho je docíleno globálním mapováním. Globální mapování je zde popsáno rovnicí 3.1

$$g(x, y) = L + f(\theta) \cdot C, \quad (3.1)$$

kde g je výsledný odstín bodu na souřadnicích (x, y) , L je světlost vstupního bodu v barevném modelu LCH, C je složka C v tomto modelu a θ odpovídá úhlu H (z modelu LCH). Ústřední problémem metody je tedy najít funkci f , která bude mapovat úhel θ modelu LCH na míru zastoupení sytosti ve výsledném odstínu. Funkce f je definována takto:

$$f(\theta) = \sum_{k=1}^n (A_k \cos(k\theta) + B_k \sin(k\theta)) + A_0. \quad (3.2)$$

V rovnici 3.2 jsou A_k , B_k a A_0 neznámé parametry, a problémem tedy je jejich nalezení. Zde bohužel nelze s výkladem pokračovat přímo chronologicky; hodnoty A_k , B_k a A_0 budou zmíněny až v rovnici 3.7.

Funkce energie, gradientu a rozdíl barev jsou definovány takto:

$$E_s = \sum_{(x,y) \in \omega} \|\nabla g(x, y) - G(x, y)\|^2, \quad (3.3)$$

$$\nabla g(x, y) = (g(x+1, y) - g(x-1, y), g(x, y+1) - g(x, y-1))$$

$$G(x, y) = \begin{pmatrix} G^x(x, y) \\ G^y(x, y) \end{pmatrix}^T = \begin{pmatrix} c(x+1, y) \ominus c(x-1, y) \\ c(x, y+1) \ominus c(x, y-1) \end{pmatrix}^T$$

Pro následnou implementaci je podstatné, že pro výpočet gradientu se používají centrální diference.

Jedním z klíčových prvků této metody je způsob, jakým se zde počítá barevný rozdíl. Ten je definován následujícím vztahem 3.4.

$$c_i \ominus c_j = \text{sign}(c_i, c_j) \sqrt{\Delta L_{i,j}^2 + \left(\alpha \frac{\sqrt{\Delta a_{ij}^2 + \Delta b_{ij}^2}}{\mathcal{R}} \right)^2} \quad (3.4)$$

Kde \mathcal{R} je konstanta o velikosti $\mathcal{R} = 2.54\sqrt{2}$. Rovnici 3.2 lze vektorově zapsat takto:

$$f(\theta) = t^T x, \quad (3.5)$$

což je redukovaný zápis rovnice 3.3. V rovnici 3.5 jsou t a x vektory o rozměrech $(2n+1) \times 1$. $t = (t_i)$, $x = (x_i)$. Tyto vektory jsou definovány následovně:

$$t_i = \begin{cases} \cos(i\theta) & \text{pro } 1 \leq i \leq n \quad (i = \{1, 2, 3, 4\}) \\ \sin((i-n) \cdot \theta) & \text{pro } n+1 \leq i \leq 2n \quad (i = \{5, 6, 7, 8\}) \\ 1 & \text{pro } i = 2n+1 \quad (i = \{9\}) \end{cases} \quad (3.6)$$

$$x_i = \begin{cases} A_i & \text{pro } 1 \leq i \leq n \quad (i = \{1, 2, 3, 4\}) \\ B_{i-n} & \text{pro } n+1 \leq i \leq 2n \quad (i = \{5, 6, 7, 8\}) \\ A_0 & \text{pro } i = 2n+1 \quad (i = \{9\}) \end{cases} \quad (3.7)$$

Množiny v závorce odpovídají hodnotám i pro $n = 4$. Tato hodnota byla autory metody označena jako prokazatelně dobrá volba.

Pro spočítání M_s a b_s je třeba znát vektory u a v a skaláry p a q . Vektory u a v jsou definovány takto:

$$u = (C \cdot t)_x, \quad (3.8)$$

$$v = (C \cdot t)_y. \quad (3.9)$$

Vektor t byl již dříve definován v rovnici 3.6. Dolní indexy x a y označují parciální derivace v tomto směru. Na místo hodnoty C je tedy dosazováno ve směru x $C(x-1, y) - C(x+1, y)$ atd. Stejně jako výše, písmeno C zde označuje hodnotu C barevného prostoru LCH vstupního obrázku. Výpočet vektoru v je pak velmi podobný kódu uvedeném ve výpise 3.1, protože se také zakládá na násobení vektorem t (3.6). Výpočet hodnot p a q je pak už jednoduchý:

$$p = G^x - L_x, \quad (3.10)$$

$$q = G^y - L_y. \quad (3.11)$$

Je ale opět ještě třeba vyjádřit G^x , G^y , L_x , a L_y .

G^x a G^y označují gradient ve směru určeném horním indexem. Výpočet gradientu u této metody byl již dříve vyjádřen v rovnici 3.4.

L_x a L_y jsou již obyčejné centrální difference na ose určené dolním indexem, počítané z jasové složky.

Jakmile jsou známy klíčové vektory u a v a skaláry p a q , je možné se posunout k vyjádření matice M_s a vektoru b_s . Jejich zápis vypadá takto:

$$M_s = \sum_{\Omega} (uu^T + vv^T), \quad (3.12)$$

$$b_s = \sum_{\Omega} (pu + qv). \quad (3.13)$$

V rovnici 3.12 horní index T označuje transpozici vektoru. Cílem celé této snahy bylo právě získat naplněnou matici M_s a vektor b_s . Z těchto hodnot již lze totiž snadno vyjádřit vektor \hat{x}_{image} , který jednoznačně definuje funkci f zmíněnou v rovnici 3.1 a tím také přesně určuje výsledné nelineární mapování. Matice \hat{x}_{image} se tedy získá řešením této rovnice:

$$\hat{x}_{image} = (M_s + \lambda I)^{-1} b_s.$$

Horní index -1 zde označuje pseudoinverzní matici. V kapitole 2.3.2 se pseudoinverzní matice značila A^+ , protože je to v matematice častější označení, nicméně zde je použito značení A^{-1} , protože se to takto používá v [18].

Matici \hat{x}_{image} je možné přímo použít do rovnice 3.5 a získat tak funkci f .

Rovnice 3.4 definovala to, jak se v této metodě počítá barevný rozdíl. Součástí rovnice je také funkce *sign*, která zatím nebyla popsána. Funkce *sign* pracuje v následujících třech krocích:

1. Použití znaménka z funkce H-K efekt, přesněji z ∇L^{HK} . Používá se zde varianta Nayatani VAC. Tento algoritmus byl podrobně popsán v kapitole 2.3.6 a funguje pro řazení barev s velkou úspěšností.
2. Znaménko ΔL , kde L je jasová složka v barevném modelu LCH.

3. Znaménko $\Delta L^3 + \Delta a^{*3} + b^{*3}$, kde L , a^* a b^* jsou složky z barevného modelu $L^*a^*b^*$.

Tyto kroky jsou seřazeny podle priority od nejvyšší po nejnižší a jejich výpočet probíhá sekvenčně. Jakmile je hodnota v nějakém kroku rovna 0, nelze použít znaménko z tohoto kroku a posouvá se na krok s nižší prioritou. Jakmile se narazí na nenulovou hodnotu, hledání znaménka se přerušuje a na nižší úroveň se již neposouvá (toto by mělo nastat ve většině případů hned v kroku 1).

Vlastní implementace

Implementace metody Kim09 byla rozdělena do šesti metod. Drobné komplikace nastaly z důvodu použití knihovny `matrix.h`. Tato knihovna je šířena v jednom hlavičkovém souboru, který obsahuje také těla funkcí. Z tohoto důvodu nemůže být vložena do hlavičkového souboru metody `TMOKim09.h`, protože by to způsobilo chybu při linkování programu. Toto chování bylo bez nalezení řešení konzultováno také s autorem knihovny. Z toho důvodu nelze typy definované v `matrix.h` použít jako typy parametrů funkcí. Proto byla poměrně velká část kódu umístěna do metody `Transform`.

Tento operátor má pouze jediný parametr typu `double` s názvem `alpha`. V `TMOKim09.h` je možné nastavit stupeň volnosti, a to s pomocí makra `N`. To ovlivňuje nejen kvalitu výsledku, ale také jeho rychlost. V této implementaci byla použita hodnota 4, stejně jako v článku [18]. Implementace se tedy skládá z těchto metod:

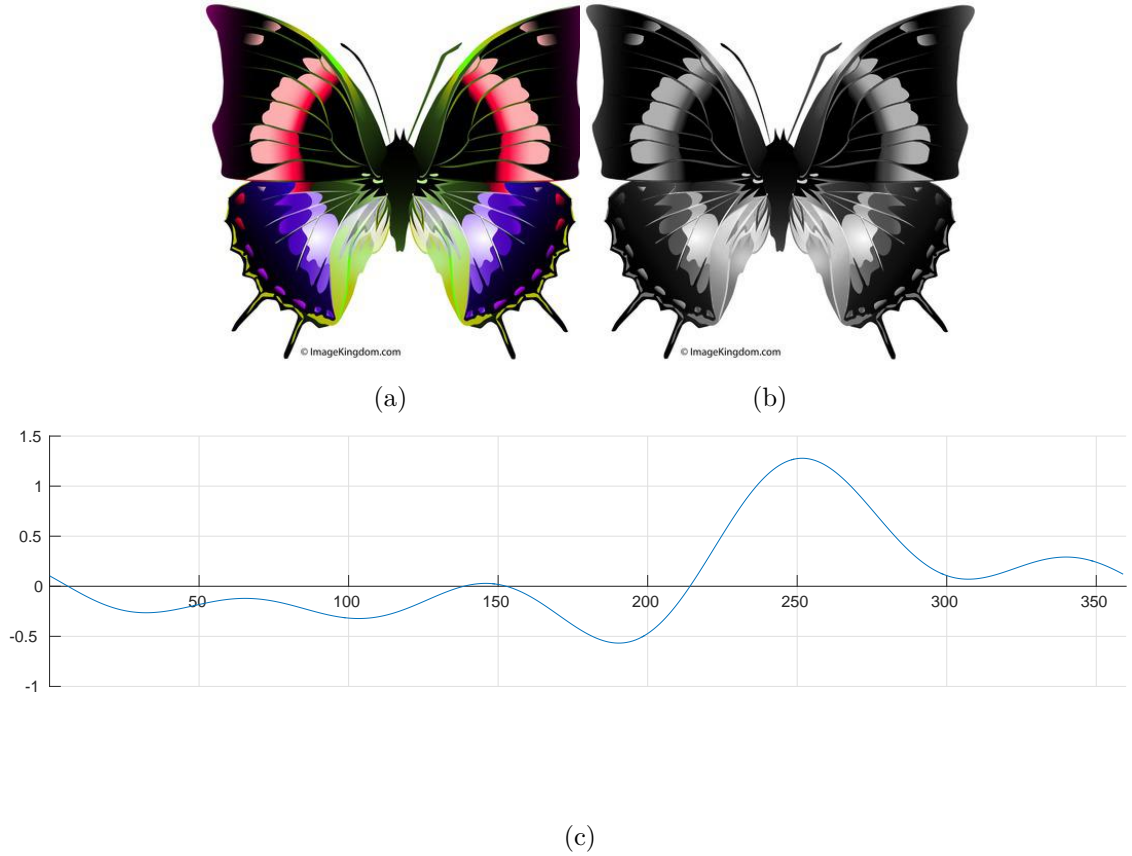
- `double Gradient(double*, double*)` – Tato metoda počítá barevný rozdíl dvou bodů. Její tělo tedy přibližně odpovídá rovnici 3.4. Metoda je volána z metody `Transform`, kde se používá pro naplnění proměnných `Gx` a `Gy`. Metoda má dva parametry typu `double *` (očekávají se trojice doublů), odpovídající operandům c_i a c_j ze zmíněné rovnice. V této metodě se také musí stanovit znaménko tohoto barevného rozdílu. Tato činnost přesně odpovídá třem krokům popsaným v 3.1.1.
- `double FunctionF(double theta)` – Toto je implementace funkce f z rovnice 3.1. Tato funkce tvoří základ nelineárního mapování, a proto je její hlavní část ve výpisu 3.1.

```

1 | for (int k-= 1; k~<= (2*N + 1); k++){
2 |     if (k~<= N){
3 |         result += cos(k * h) * X[k-1]; // k-= 1, 2, 3, 4
4 |     }else if ((k~> N) && (k~< 2*N + 1)){ // k-= 5, 6, 7, 8
5 |         result += sin((k - N) * h) * X[k-1];
6 |     }else{ // k-= 9
7 |         result += X[k-1];
8 |     }
9 | }
```

Výpis 3.1: Výpočet hodnoty funkce f

Tento kód přesně odpovídá polynomiálnímu zápisu 3.2 s hodnotami A_k , B_k a A_0 určenými podle rovnice 3.7. Průběh této mapovací funkce může vypadat např. tak, jak je naznačeno na obrázku 3.2c. Na ose X jsou hodnoty v rozsahu $(0, 360)$, protože na této ose jsou odstíny – složka H barevného modelu LCH . Tato funkce tedy definuje míru vlivu jednotlivých odstínů ve výstupním obrázku.



Obrázek 3.2: Ukázka možného průběhu funkce f (3.2c). Zde zobrazena spolu s analyzovaným obrázkem 3.2a a již převedeným obrázkem 3.2b.

- `double HkEffectPredictor(double *)` – Tato funkce počítá Helmholtz-Kohlrausch (zkracováno HK) prediktor popsany v 2.3.6. Pro činnost této funkce je nutné dodat referenční hodnoty bílého bodu v barevném modelu Luv. Ty jsou definované v souboru `TM0Image.h`. Dále je zde třeba nastavit hodnotu adaptivního osvětlení. V této implementaci byla zvolena hodnota 20, podobně jako je tomu v [33]. Správná funkce této metody je klíčová pro správný chod celého operátoru. Bohužel variant jak HK implementovat existuje více než jedna. V dokumentu [24] je výstup prediktoru definován takto:

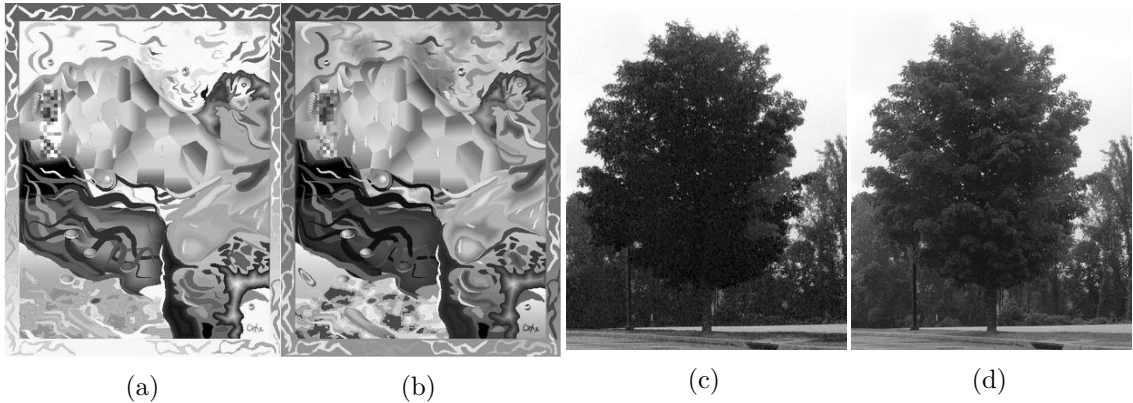
$$\Gamma_{VAC} = 1 + (-0.1340q(\theta) + 0.0872K_{Br})s_{uv}(x, y), \quad (3.14)$$

zatímco v [33] je definován takto:

$$L_{eq*} = L * + (-0.1340q(\theta) + 0.0872K_{Br})s_{uv}(x, y)L * . \quad (3.15)$$

Během implementace probíhaly experimenty s oběma variantami. Porovnáním výsledků bylo zjištěno, že varianta 3.15 produkuje podstatně lepší výsledky a proto byla v metodě nakonec použita. Toto je doloženo na obrázcích 3.3. Obrázky 3.3a a 3.3c

používají rovnici 3.14 a 3.3b a 3.3d používají 3.15. Na obázku je zřetelné, že 3.3c obsahuje méně detailů, než 3.3d a 3.3a zase obsahuje přepálené oblasti.



Obrázek 3.3: Srovnání výsledků při požití různých rovnic 3.14 a 3.15 pro výpočet HK prediktoru. 3.3a a 3.3c používají rovnici 3.14, zatímco zbývající dva obrázky používají 3.15.

Tato funkce očekává ve svém parametru trojici hodnot v barevném modelu Luv.

- `void PixelLabToLuv(double*, double*)` – Jedná se o jednoduchou obalovací metodu pro převod jednoho bodu mezi barevnými modely. Interně tato metoda volá metody z `TMOImage`. Převádění po bodech je výhodné, protože není třeba mít uložený celý obrázek v paměti několikrát jen s různým barevným modelem.
- `void PixelLchToLab(double*, double*)` – Podobně jako výše, jde o jednoduchou funkci pro převodu bodu z jednoho barevného modelu do druhého.
- `int Transform()` `Transform` je základní převodová funkce každé ze tří implementovaných metod. V tomto operátoru je z důvodů popsaných výše poměrně obsáhlá. Tato funkce by se dala poměrně jednoduše rozdělit do dvou logických kroků. Prvním krokem je analýza obrázku, naplnění všech potřebných proměnných a hlavně matic. V druhém kroku pak probíhá na základě těchto dat samotné převádění. A skutečně, v této metodě je třeba dvou průchodů celým obrázkem. V prvním průchodu se nejprve plní pomocné proměnné s daty z okolí zkoumaného bodu takto:

```
1 || c_shift_left = (i== 0) ? pSrc->GetPixel(i, j)[1] : pSrc->GetPixel(i -
   || 1, j)[1];
2 || c_shift_right = (i== pSrc->GetWidth() - 1) ? pSrc->GetPixel(i, j)[1] :
   || pSrc->GetPixel(i + 1, j)[1];
```

Výpis 3.2: Vyjádření pomocných proměnných pro snadnější práci s okolím bodu

Těchto proměnných je celkem 8, jsou vyjádřeny pro každý směr pro komponenty c a h . Použití těchto pomocných proměnných ve výsledku velmi ulehčí práci a zpřehlední kód, protože proměnné se používají mnohokrát a bez nich by byl kód významně delší. Dále jsou vypočteny vektory u a v podle rovnic 3.8 a 3.9. Dále jsou s pomocí metody `Gradient` naplněny proměnné Gx a Gy a vyjádřeny jsou také hodnoty Lx a Ly . Z těchto 4 hodnot se vyjádří hodnoty p a q podle rovnic 3.10 a 3.11. V tomto bodě jsou již známy všechny hodnoty pro výpočet matice Ms a vektoru bs podle rovnic 3.12

a 3.13. Všechny kroky popsané v tomto odstavci probíhají v cyklu, který běží přes celý obrázek a mají za úkol obrázek ohodnotit a připravit data pro tvorbu funkce f .

Po dokončení tohoto procesu je již snadné naplnit matici X a tím jsou připravena všechna data pro funkci f . Druhý průchod obrázkem je již velmi rychlý a probíhá v něm samotné nelineární mapování, které je naprogramováno takto:

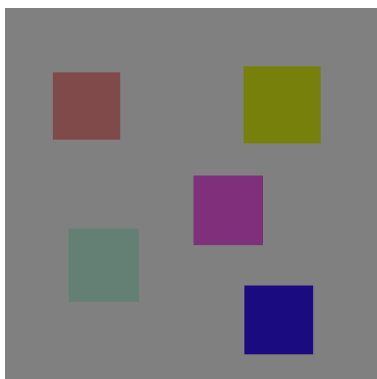
```
1 || g = L + FunctionF(h)*c;
```

Výpis 3.3: Finální výpočet odstínu jednoho bodu

I přes drobné komplikace byla práce na této metodě velmi usnadněna použitím knihovny [17]. Díky tomu nebylo potřeba např. při násobení matic nebo inverzi matic používat cykly, ale stačilo zavolat metodu.

3.1.2 Efficient Color-to-Gray Conversion for Digital Images in Gradinet Domain

Efficient Color-to-Gray Conversion for Digital Images in Gradinet Domain [32] je metoda převodu barevných obrázků na černobílé v gradientní oblasti. Z gradientů je poté zpět rekonstruován černobílý obrázek za použití takzvaného PES (Poisson Equation Solver). Tato metoda si bere za cíl shodu intenzit vstupního a převedeného obrázku na základě vnímání osvětlení lidským okem, přičemž barvy v isoluminantních obrázcích³ by měly být převedeny na různé odstíny, aby nesplývaly v jednu barvu ale aby bylo možné mezi nimi rozlišovat. Další z vlastností této metody je, že zachovává významné prvky obrázku. Příklad problému při převodu isoluminantních obrázků je vidět na obrázku 3.4. Při tomto převodu byla jednoduše použita složka jas jako odstín výsledného černobílého obrázku.



(a) Isoluminantní vstupní obrázek.



(b) Převedený černobílý obrázek.

Obrázek 3.4: Ukázka problému při „naivním“ převádění isoluminantních obrázků.

Princip činnosti

Tato metoda pracuje výhradně v barevném modelu CIELAB (2.1.3). Vstupní obrázek je tedy reprezentován 2D maticí trojic $L(x, y)$, $A(x, y)$, $B(x, y)$. Cílem je získat gradientní

³Isoluminantní obrázky jsou obrázky s konstantní intenzitou, ale přesto různou barvou. Mohou být uloženy v libovolném barevném modelu.

pole $\tilde{\nabla}C$, které bude poté vloženo na vstup PES solveru pro rekonstrukci černobílého obrázku. Toto gradientní pole se skládá ze dvou komponent. Těmi jsou gradient světlosti ∇L a gradient barevnosti $\tilde{\nabla}c(a, b)$. ∇L se podle [32] získá pomocí vztahu:

$$\nabla L = \nabla I = (I_x, I_y) = (I(x + \Delta x, y) - I(x, y), I(x, y + \Delta x) - I(x, y)),$$

$I(x, y)$ označuje intenzitu (světlost) ve vstupním obrázku na souřadnicích (x, y) . Také je třeba dodržet znaménka u Δx , protože při jejich zaměnění to vede k inverznímu obrázku (negativu). Druhá komponenta $\tilde{\nabla}c(a, b)$ se vypočítá takto:

$$\tilde{\nabla}c(a, b) = (C_x, C_y)$$

$$C_x = \sqrt{(a(x + \Delta x, y) - a(x, y))^2 + (b(x + \Delta x, y) - b(x, y))^2} \quad (3.16)$$

$$C_y = \sqrt{(a(x, y + \Delta y) - a(x, y))^2 + (b(x, y + \Delta y) - b(x, y))^2} \quad (3.17)$$

Když jsou známé komponenty ∇L a $\tilde{\nabla}c(a, b)$, je možné spočítat $\tilde{\nabla}C$:

$$\begin{aligned} \tilde{\nabla}C = & (sign(L_x, a(x + \Delta x, y), a(x, y), b(x + \Delta x, y), b(x, y)) \cdot \sqrt{L_x^2 + A^2(C_x)}, \\ & sign(L_y, a(x, y + \Delta y), a(x, y), b(x, y + \Delta y), b(x, y)) \cdot \sqrt{L_y^2 + A^2(C_y)}) \end{aligned}$$

V této rovnici byla použita funkce A . Jedná se o tzv. *attenuační* (tlumící) funkci. Její výpočet probíhá takto:

$$A(x) = x \cdot \left(\beta \left(1 - \left(\frac{x}{cx_{max}} \right)^\gamma \right) \right) = x \cdot A_0(x) \quad (3.18)$$

Hodnoty β a γ jsou parametry tohoto operátoru. c je konstanta zajišťující, že nejvyšší hodnoty barevného rozdílu nebudou plně utlumeny. Funkce 3.18 je důležitá pro omezení zkreslení, které by bez ní mohlo při převodu barevného obrázku na černobílý vzniknout. Zbývá ještě vyjádřit funkci $sign$. Správné pořadí barev je typický problém u isoluminantních obrázků. Právě k řešení tohoto problému slouží funkce $sign$, která k výpočtu využívá informace o barevnosti:

$$sign(\Delta L, a_2, a_1, b_2, b_1) = sign(\Delta L + \alpha \cdot (\vec{v}_\theta \cdot \vec{\Delta}_C)), \quad (3.19)$$

kde L_1 , a_1 a b_1 jsou informace o první barvě a L_2 , a_2 a b_2 jsou informace o druhé barvě. $\Delta L = L_1 - L_2$. \vec{v}_θ a $\vec{\Delta}_C$ jsou vektory o rozměrech 2×1 . $\vec{v}_\theta = (\cos(\theta), \sin(\theta))$ a $\vec{\Delta}_C = (a_2 - a_1, b_2 - b_1)$. Tento vektor se nazývá vektor chromatické vzdálenosti.

Tato metoda také definuje *modulovaný barevný rozdíl*. Tento barevný rozdíl pracuje s tlumící funkcí a tím minimalizuje možné zkreslení. Je definován takto:

$$\Delta E = \sqrt{(\Delta L)^2 + (A(\sqrt{(\Delta a)^2 + \Delta b^2}))^2}$$

Jak je vidět v rovnicích 3.18 a 3.19, jejich výsledek je silně závislý na hodnotách α , β , γ , θ . To jsou nastavitelné parametry a přímo odpovídají parametrům celé metody. Jejich rozsahy jsou: $\alpha \in [0, 1]$, $\beta \in [0, \infty]$, $\gamma \in (0, \infty)$ a $\theta \in [0, 2\pi]$. V tlumící funkci se používají parametry β a γ . Parametr γ ovlivňuje míru zachování vysokých barevných rozdílů. Vysoké γ (až ∞) znamená zachování velkých barevných rozdílů, zatímco nízké γ tyto rozdíly utlumí. Parametr β slouží pro škálování výsledků tlumící funkce. Význam těchto dvou

parametrů je v tom, že pomocí jejich vhodného nastavení lze docílit redukce barevných artefaktů, které by vznikly bez tlumicí funkce. Parametry α a θ se používají k určení znaménka u isoluminantních barev. Parametr α význam barevného rozdílu při určování výsledného znaménka. Respektive ovlivňuje jeho sílu v poměru k síle rozdílu intenzit osvětlení. Hodnota 0 by znamenala, že výsledné znaménko bude závislé pouze na ΔL (rozdíl intenzit osvětlení). Maximální hodnota 1 pak znamená, že ΔL i barevný rozdíl budou mít stejnou váhu. Posledním parametrem je θ . Tento parametr určuje úhel pro výpočet znaménka v rovině a-b prostoru CIELAB. Pro snadnější manipulaci se hodnota tohoto parametru zadává ve stupních, nikoli v radiánech, tedy v rozsahu $[0, 360)$.

Tímto postupem je tedy vypočítáno gradientní pole $\tilde{\nabla}C$. Pro rekonstrukci černobílého obrázku je nutné použít PES solver, protože toto gradientní pole nemusí být konzistentní. Během implementace bylo provedeno testování, při kterém se černobílý obrázek získával metodou prosté integrace, tedy bez PES solveru. Obrázky získané touto metodou jsou na obrázku 3.5. Nejprve vznikl obrázek 3.5a. Ten vznikl prostou integrací ve směru osy X. Integrace probíhala zleva doprava. Charakteristické pro tento obrázek je, že první hodnoty v řádku (tedy hodnoty v levém sloupci obrázku) vznikly přímo z prvního gradientu. Odstín celého řádku je tedy přímo závislý na hodnotě v prvním sloupci — právě to tvoří vodorovné pruhy. Toto je samozřejmě špatně a v obrázku 3.5b byla snaha tento efekt eliminovat. Bylo tedy třeba zjistit počáteční hodnotu v levém sloupci, ze které bude možno vycházet, a integrace se bude od ní odvíjet. V tomto obrázku se pro počáteční hodnotu v řádku použila hodnota L ze zdrojového obrázku. Ve výsledku je potom vidět, že ze začátku (v levé části obrázku) tento postup funguje a obrázek se na svém levém okraji jeví bez chyb. Čím více se však integrace posouvala vpravo, tím více se projevily modifikace gradientního pole (které způsobují jeho nekonzistentnost), a tím více jsou výrazné vodorovné pruhy. Na posledním obrázku 3.5c byla tato metoda ještě zdokonalena a obrázek byl integrován nejprve ve směru osy X, a poté i ve směru osy Y. Zlepšení to ale podle očekávání přineslo jen malé. Také zde se projevila nekonzistentnost gradientního pole, která způsobila svislé i vodorovné pruhy. To je tedy důvod, proč je nezbytné použít pro rekonstrukci PES solver. Na závěr byly všechny obrázky normalizovány do rozsahu $[0, 255)$.

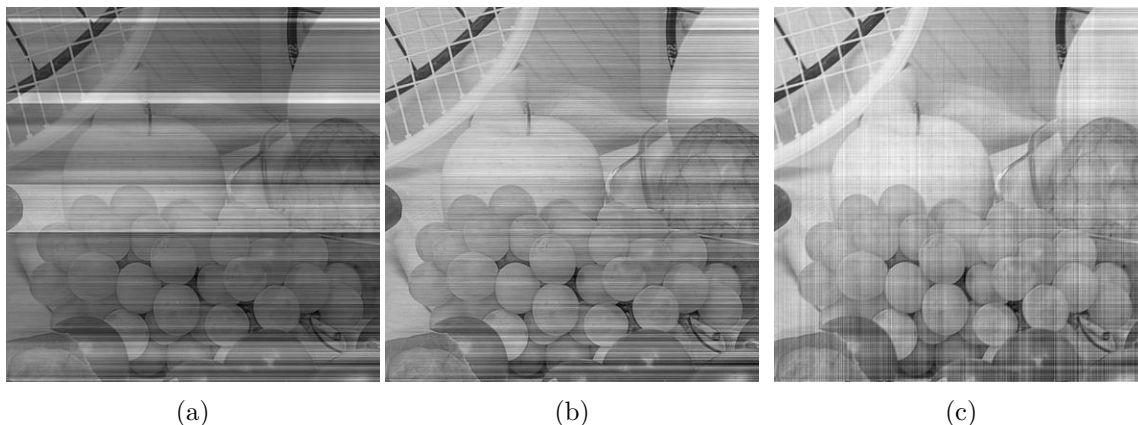
Metoda prosté integrace by samozřejmě fungovala s téměř nulovou chybou při použití klasického gradientu. Tento fakt byl během práce ověřen experimentem, při kterém bylo gradientní pole vytvořeno pouze ze složek L obrázku. Toto gradientní pole bylo konzistentní, a použití PES solveru tedy bylo zbytečné.

Vlastní implementace

Chod tohoto operátoru lze rozdělit do dvou kroků: konstrukce gradientního pole $\tilde{\nabla}C$ a jeho zpětná rekonstrukce do černobílého obrázku za pomoci Poisson equation solveru (2.3.5). PES nebyl v rámci této práce implementován, ale byl použit solver z 3.4.2.

V hlavičkovém souboru `TMOZheng15.h` je definováno makro `CONSTANT_C`, které je použito v tlumicí funkci 3.18. V implementaci pro tento projekt byla pro toto makro zvolena hodnota 2, což odpovídá hodnotě ze článku [32]. Dále se operátor skládá z těchto metod:

- `double AttenuationFunction(double)` – Jednoduchá tlumicí funkce, její tělo odpovídá rovnici 3.18. Používají se zde parametry operátoru β a γ a také makro `CONSTANT_C`.
- `double Sign(double, double, double, double, double, double)` – Znaménková funkce pro určení orientace gradientu, vychází z rovnice 3.19. V této funkci probíhá násobení



Obrázek 3.5: Srovnání různých druhů rekonstrukce obrázku z gradientu bez použití PES solveru.

vektorů \vec{v}_θ a $\vec{\Delta}_C$. Tyto vektory jsou však malé, že není třeba používat žádný speciální datový typ a toto násobení je implementováno klasickým násobením skalárů.

- `double ChromaticGradientComponent(double, double, double, double)` – Funkce pro výpočet barevné komponenty gradientu. Její tělo se shoduje s rovnicemi 3.16 a 3.17. Díky tomu, že jsou difference spočítány již v invokující metodě, je definice této funkce velmi jednoduchá.
- `double GradientFiledComponent(double, double, double, double, double, double)` – Metoda počítající celý jeden ze dvou komponentů (pro směr x a y) v gradientním poli. Tato metoda je matematicky popsána v rovnici 3.18. V této funkci je nejzajímavější tento řádek:

```
1 || return (result != result) ? 0.0 : result;
```

Výpis 3.4: Ošetření hodnoty NaN v gradientní poli.

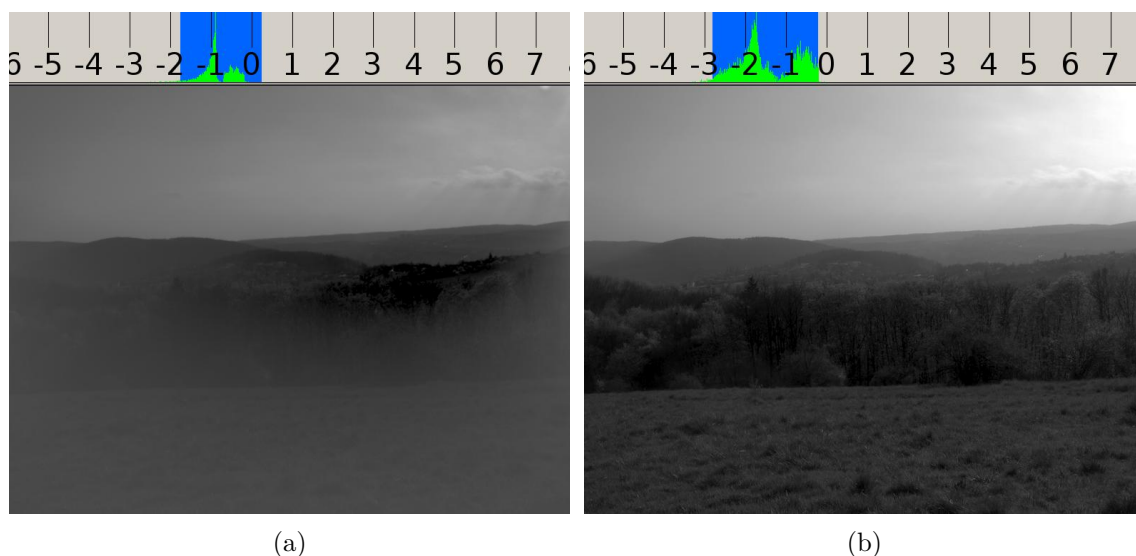
Řádek 3.4 ošetřuje, aby se do gradientního pole nedostala hodnota Nan. Ta do něj mohla být zanesena některou z funkcí popsaných výše. Bez tohoto ošetření metoda fungovala správně pro přibližně 80 % testovací sady. U zbytku byl převedený obrázek celý černý, protože použitý solver nedokázal obrázek rekonstruovat.

- `int Transform()` – V transformační metodě se sestaví gradientní pole $\vec{\nabla}C$ s pomocí výše popsaných metod. Toto pole je uloženo do struktury složené z dvou složek typu `Matrix` z knihovny `matrix.h` (3.4.1), každá je zde pro jeden směr. Pro správnou funkci tohoto operátoru je důležité použití korektního druhu diferencí. Při první derivaci je třeba použít dopředné difference (2.3) a při druhé derivaci je třeba použít zpětné difference (2.4). Použití centrálních diferencí (2.5) v kterékoli z těchto dvou derivací vede na neostré obrázky. K tomuto výsledku vede také použití stejného typu diferencí u obou derivací, tedy dopředná-dopředná, nebo zpětná-zpětná. Použití opačných diferencí, ale v opačném pořadí než bylo uvedeno v úvodu tohoto odstavce, tedy v pořadí zpětná-dopředná vede také k dobrým výsledkům – ostrým obrázkům, ale obrázky jsou o něco více jasné, než by měly být. Tento jev je ilustrován na obrázku 3.6.



Obrázek 3.6: Srovnání výsledných obrázků při použití centrálních diferencí 3.6a a při použití diferencí v pořadí dopředná-zpětná 3.6b. Při detailním pohledu je vidět, že hrany v obrázku 3.6a nejsou ostré.

Tento problém se v ještě daleko větší míře projeví u HDR obrázků, kde při nesprávném použití diferencí dochází k silnému vyblednutí a také ztrátě dynamického rozsahu. Toto je ukázáno na obrázku 3.7. Histogramy nad obrázky budou vysvětleny v kapitole 4.1.



Obrázek 3.7: Dopad nevhodného použití diferencí při převodu HDR obrázků metodou Zheng15. Počítání s pomocí centrálních diferencí 3.7a a výpočet pomocí zpětných diferencí 3.7b.

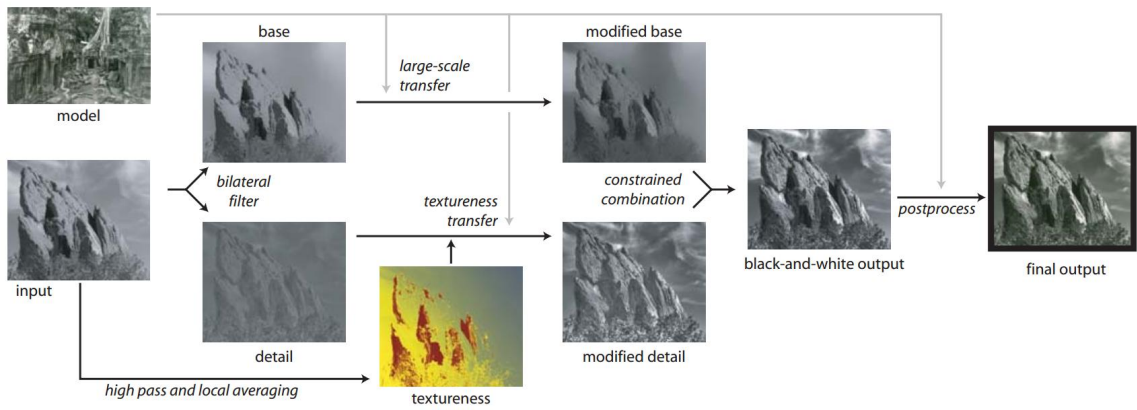
Jakmile je sestaveno gradientní pole, je zavolána metoda `pde::poisolve` z knihovny `poisson_pde` pro rekonstrukci černobílého obrázku. Před uložením obrázku se ještě provede jeho převedení do požadovaného rozsahu a normalizace.

3.1.3 Two-scale Tone Management for Photographic Look

Tato metoda [3] byla prezentována v roce 2006 na konferenci SIGGRAPH. To je významná světová konference s dlouhou tradicí zaměřující se na počítačovou grafiku. Metoda je založena na technice histogram matching, ale sofistikovaně pracuje s detaily a přináší tak kvalitnější výsledky. Množství detailů ve výsledném obrázku je dokonce možné nastavit. Účelem metody je tedy přenést fotografický vzhled z jednoho obrázku na druhý.

Globální kontrast

Základní myšlenou této metody je rozdělení zpracování na dvě části počítané odděleně. Tyto části se zde nazývají *base* a *detail* a přenášejí v tomto pořadí globální a lokální kontrast. Schéma činnosti této metody je na obrázku 3.8.



Obrázek 3.8: Znázornění jednotlivých kroků metody Two-scale Tone Management for Photographic Look. Obrázek převzat z [3].

Komponenta *base* bude označována jako B a získá se z vstupního obrázku I s pomocí bilaterálního filtru (2.8):

$$B = bf(I)$$

Pro správnou funkci je kritické správné nastavení parametrů filtru σ_s a σ_r . Jak již bylo popsáno v kapitole 2.8, tyto parametry označují prostorovou váhu a váhu rozsahu tohoto filtru v tomto pořadí. Parametr σ_s se získá pomocí tohoto vztahu:

$$\sigma_s = \min(\text{sirka_vstupniho_obrazku}, \text{vyska_vstupniho_obrazku})/16 \quad (3.20)$$

Parametr σ_r se pak vypočítá takto:

$$\sigma_r = p_{90}(\|\nabla I\|),$$

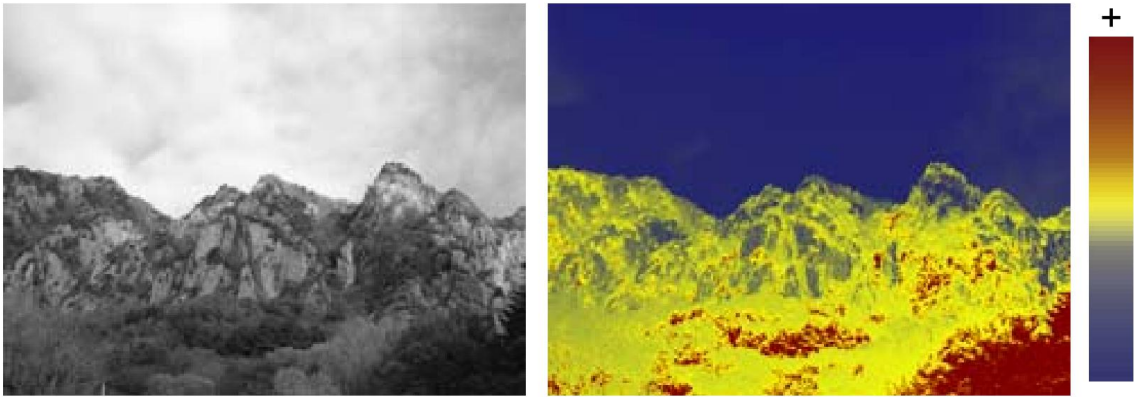
$p_{90}(I)$ značí percentil 90 % z hodnot obrázku I . Dále je stejným způsobem získán *base* modelového obrázku B_M . Pomocí histogram matching se přenesou histogram *base* modelu B_M do nového *base* B_I .

Lokální kontrast

Lokální kontrast se zde stará o přenos detailů obrázku. Složka detail se získá prostým odečtením base B od původního obrázku I:

$$D = I - B.$$

Může se tedy stát, že některé pixely složky detail budou mít zápornou hodnotu. Ze složky detail je dále potřeba vypočítat „vrstvu“ nazývanou *textureness*. Tato složka vyjadřuje lokální množství detailů v jednotlivých místech obrázku. Toto je znázorněno na obrázku 3.9.



Obrázek 3.9: Vstupní obrázek (vlevo) a jeho *textureness* - lokální množství detailů (vpravo). Obrázek převzat z [3].

Výpočet *textureness* probíhá podobně jako u bilaterálního filtru[3]:

$$T(I)_p = \frac{1}{k} \sum_{q \in |H|} g_{\sigma_s}(\|p - q\|) g_{\sigma_r}(|I_p - I_q|) |H|_q,$$

kde:

$$k = \sum_{q \in I} g_{\sigma_s}(\|p - q\|) g_{\sigma_r}(|I_p - I_q|)$$

V podstatě se tedy jedná o upravený bilaterální filtr, který „kříží“ $|H|$ a I , protože pro výpočet jmenovatele se nepoužívá $|H|$ jak by tomu bylo u klasického bilaterálního filtru, ale I . Tento filtr má na vstupu $|H|$. $|H|$ označuje absolutní hodnotu z výstupu horní propusti. Při výpočtu se také používají parametry σ_s a σ_r označující rozptyl. Hodnota σ_r zůstává stejná jako u klasického bilaterálního filtru pro výpočet base a její výpočet je popsán v kapitole 3.1.3. Rozptyl σ_r je zde vynásoben konstantou 8. Díky tomu umí tento filtr detekovat hrany s větším kontrastním rozdílem.

Horní propust sama o sobě zvýrazňuje hrany. Bilaterální filtr počítá lokální vážené průměry v obrázku. Protože lokální průměr horní propusti je nula, je potřeba použít absolutní hodnotu z ní.

Pro samotný přenos *textureness* je třeba vyjádřit poměr ρ . Jedná se opět o pole o rozměrech původního obrázku. Tento výpočet se udělá podle tohoto vztahu[3]:

$$\rho_p = \max\left(0, \frac{T'_p - T(B')_p}{T(D)_p}\right). \quad (3.21)$$

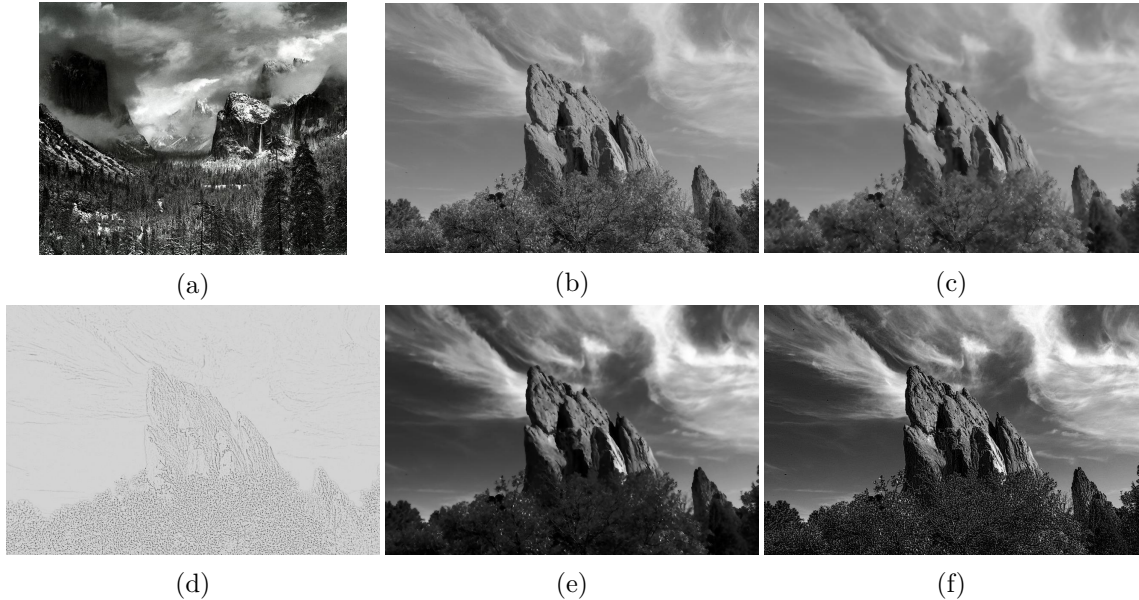
V této rovnici T' značí výsledný (cílený) textureness. T' se získal pomocí histogram matching z $T(M)$ do $T(I)$. $T(B')$ je textureness vypočtený z base a $T(D)$ textureness vypočtený z detailu. Dolní index p označuje, že tento výpočet je třeba provést pro každý bod obrázku. Čitatel tohoto zlomku je jakousi obdobou rovnice $D = I - B$ a udává tedy jakýsi detail, ale mezi T a $T(B)$. Pokud se tento detail vydělí $T(D)$, získá se poměr těchto detailů, který určuje množství detailů do výsledného obrázku. Pro zabránění obrácení gradientu se počítá pouze s kladnými hodnotami, což zajišťuje funkce \max .

Výsledný odstín se pak vypočítá takto:

$$O = B' + \rho D$$

Kde B' je base po provedení histogram matching. Protože ρ může nabývat vysokých hodnot, je třeba dbát na to, aby hodnoty O nepřetékly mimo zobrazitelný rozsah. Pokud by k tomu došlo, je třeba provést kompresi histogramu.

Činnost tohoto operátoru je stručně naznačena na obrázku 3.10. První obrázek 3.10a je model, 3.10b pak představuje vstupní obrázek. Na obrázku 3.10c je base vypočtená ze vstupního obrázku, silný úbytek detailů je patrný hlavně na stromech v popředí. Na základě ní byla určena vrstva detailů na obrázku 3.10d. Obrázek 3.10e zobrazuje base, na který byl aplikován histogram matching. Poslední obrázek 3.10f pak kombinuje obrázky 3.10e a 3.10d. Je na něm base upravený technikou histogram matching spolu s detaily násobenými hodnotou $\rho = 3$. Ve skutečné implementaci zde však samozřejmě není konstanta, ale ρ se spočítá jako podíl zapsaný v rovnici 3.21.



Obrázek 3.10: Zjednodušený přehled nejdůležitějších kroků činnosti operátoru Bae06.

Obrázek 3.10d byl pro větší názornost invertován (ve skutečnosti je naprostá většina obrázku černá – bez detailu) a byl mu upraven kontrast. Hodnoty detailu se pohybují v tak nízkém rozsahu, že by na tomto obrázku bez této úpravy nebylo nic rozpoznatelné.

Vlastní implementace

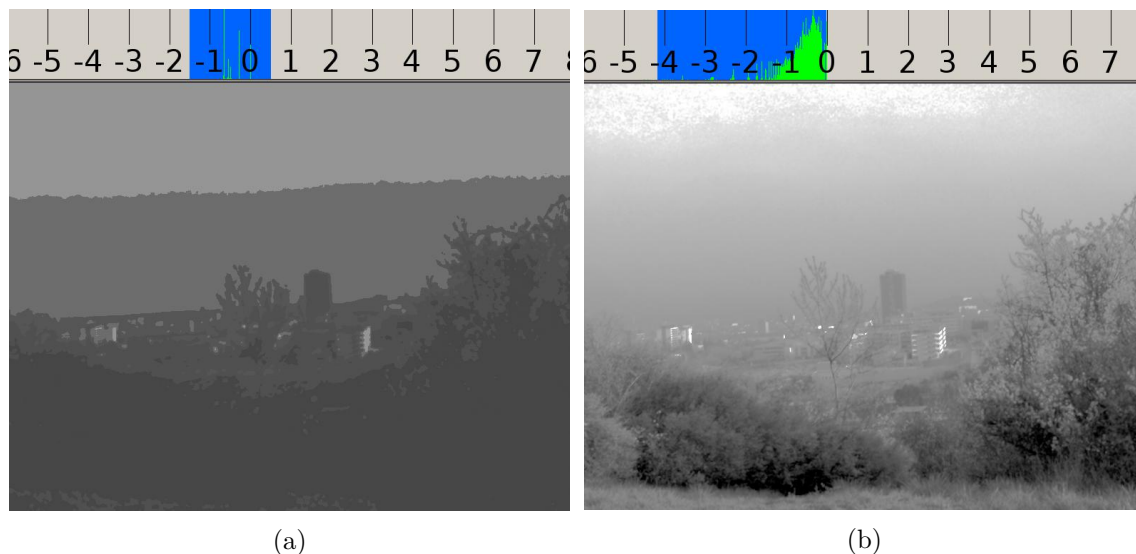
Tento operátor je mírně složitější než předcházející dvě metody, protože není zaměřen pouze na převod barevných obrázků na černobílé, ale spíše na přenos vzhledu jedné fotografie na druhou. Proto se také skládá z více metod. Pro vylepšení výsledků byl tomuto operátoru přidán logický parametr „HDR“, kterým se upraví některá nastavení tak, aby operátor poskytoval lepší výsledky pro HDR obrázky. V souboru `TMOBae06.h` je např. možné makrem `HISTOGRAM_NORMALISATION` nastavit hodnotu, na kterou budou normalizovány kumulované histogramy. Ještě důležitější jsou makra `WINDOW_SIZE_LDR` a `WINDOW_SIZE_HDR` kterými se nastavuje velikost filtrovacího okna pro bilaterální filtr (2.8). Ten je sice v rovnici 2.9 definován tak, že se počítá suma přes celý obrázek, ale při praktické implementaci by takovéto řešení bylo extrémně pomalé protože pro každý bod obrázku by se musely řešit Gaussovy rovnice tolikrát, kolik má tento obrázek bodů. To by znamenalo exponenciální časovou složitost. Navíc v určité vzdálenosti od aktuálního bodu se již hodnoty Gaussovy funkce limitně blíží nule. Z těchto důvodů se používá filtrovací okno s pevnou velikostí. Dále se v `TMOBae06.h` nastavuje např. makro `MAX_RHO` určující maximální míru zesílení detailů v obrázku. Pokud by tato konstanta byla nastavena na vysokou hodnotu, vznikaly by v obrázku hrany i tam, kde předtím nebyly. Při ještě silnějším nastavení by se objevily viditelné artefakty. Zde byla zvolena hodnota 5, což odpovídá poměrně agresivnímu zvýraznění detailů. Při implementaci byly definovány tyto metody:

- `double RgbToGray(double, double, double)` – Metoda používaná dříve pro testování, později již nevyužívaná.
- `void InitialiseHistogram(int *)` – Histogramy jsou v tomto operátoru uloženy jako pole celých čísel. Tato metoda dostane parametrem ukazatel na toto pole a inicializuje jej.
- `void PrintHistogram(int *, std::string)` – Tato metoda tiskne histogram zadaný parametrem na standardní výstup. Byla používána pouze pro testování.
- `void FillHistogram(pfstmo::Array2D *, int *)` – Metoda vytvářející data histogramu. V parametru dostane vstupní obrázek a referenci na inicializovaný histogram. Počet sloupců histogramu je definován v souboru `TMOBae06.h` pomocí makra `HISTOGRAM_LEVELS`. Standardně pro LDR obrázky byla tato konstanta nastavena na hodnotu 256 s uspokojivými výsledky. Protože na počtu úrovní histogramu závisí kvalita algoritmu histogram matching a protože operátor Bae06 je na histogram matching založen, je toto velmi důležitý údaj. Hodnota 256 přinášela nepoužitelné výsledky při převodu HDR obrázků. Proto bylo do `TMOBae06.h` přidáno ještě makro `HISTOGRAM_BOOSTER`, které funguje jako jakýsi násobitel hodnoty `HISTOGRAM_LEVELS`:

```
1 || #define HISTOGRAM_LEVELS (256 * HISTOGRAM_BOOSTER)
```

Výpis 3.5: Vylepšený výpočet počtu stupňů histogramu.

V metodě `FillHistogram` je potřeba násobit intenzitu bodu také hodnotou `HISTOGRAM_BOOSTER`, protože jinak by se sice zvýšil počet zkoumaných úrovní, ale nezvýšila by se skutečná jemnost histogramu. S rostoucí hodnotou `HISTOGRAM_BOOSTER` roste také kvalita převedených obrázků, ale s tím také paměťová náročnost operátoru. Operátor je ještě dobře použitelný při `HISTOGRAM_BOOSTER` nastaveném na hodnotu 100. Srovnání převodu stejného obrázku s a bez použití `HISTOGRAM_BOOSTER` je na obrázku 3.11.



Obrázek 3.11: Ukázka zlepšení výsledků metody Bae06 při zavedení konstanty HISTOGRAM_BOOSTER. Stav bez této konstanty 3.11a a s ní 3.11b.

- `void ComputeCumulativeHistogram(int *, int *)` – Toto je jednoduchá metoda, která z histogramu na vstupu vytváří kumulovaný histogram.
- `int FindClosestShade(int, int *)` – Další z metod pro provedení histogram matchingu. Tato metoda dostane v parametrech kumulovaný histogram a hledanou hodnotu, a vrátí index nejbližšího čísla v kumulovaném histogramu k této vstupní hodnotě.
- `void NormaliseHistogram(int *, int)` – Tato metoda provádí normalizaci kumulativního histogramu tak, aby se jeho nejvyšší hodnota rovnala zadanému maximu. Aby bylo možné metodu použít i pro klasický histogram, pracuje se dvěma průchody. V prvním průchodu se najde maximum histogramu, následně se spočítá „dělitel“ a v druhém průchodu se provádí samotná normalizace.
- `void BilateralFilter(pfstmo::Array2D *, pfstmo::Array2D *)` – Tato metoda implementuje bilaterální filtr. Prvním parametrem se určí výstupní obrázek a druhým pak předá vstupní obrázek. Na rozdíl od mnoha metod popsanych v kapitole 3.1.2, tato není pouhým přepisem rovnice 2.8 do kódu. Implementace vychází z vyjádření popsaného v knize Computer Vision: Algorithms and Applications [30]. Tam je zvlášť vyjádřena váhová funkce a zvlášť zbytek filtru.
V této funkci se pracuje s velikostí filtrovacího okna, kterou je možné ovlivnit parametrem HDR.
- `double BilateralFilterWeight(double, double, double, double, pfstmo::Array2D *)` – Toto je zvlášť vyjádřená váhová funkce zmíněná v předchozím bodě. Bilaterální filtr byl implementován tak, aby pracoval „hrubou silou“, tedy bez jakýchkoli optimalizací.
- `void CreateGrayscale(pfstmo::Array2D, TMImage *)` – Tato jednoduchá funkce pouze čte data ze vstupního obrázku v typu TMImage a ukládá jej do proměnné typu pfstmo::Array2D, s kterou se pak dále pracuje.

- `void GetDetailFromBase(pfstmo::Array2D, pfstmo::Array2D, pfstmo::Array2D)` – Tato funkce počítá rozdíl vstupního obrázku a base a tím určuje vrstvu detail.
- `double ComputeSigmaS(int, int)` – Výpočet hodnoty σ_s podle rovnice 3.20. Také tato metoda funguje mírně odlišně v režimu HDR.
- `void HistogramMatching(int *, int *, pfstmo::Array2D *)` – Metoda provádějící algoritmus histogram matching. V jejích parametrech je třeba zadat kumulativní histogram vstupního obrázku, kumulativní histogram modelu a zpracovávaný obrázek. Poslední jmenovaný je touto metodou přepsán.
- `void FillRho(pfstmo::Array2D, pfstmo::Array2D, pfstmo::Array2D, pfstmo::Array2D)` – Zde probíhá výpočet hodnoty ρ podle vzorce 3.21.
- `void FillTextureness(pfstmo::Array2D *, pfstmo::Array2D *, bool)` – Univerzální metody pro naplnění pole textureness, tato metoda je v operátoru volána hned pětkrát. V jejím těle se nejprve vstupní obrázek přefiltruje hornopropustným filtrem, pak se z výsledku spočítá absolutní hodnota, a toto je následně posláno do „křížového bilaterálního filtru“, který spočítá textureness.
- `void HighPassFilterV2(pfstmo::Array2D *)` – Implementace konvolučního hornopropustného filtru. Jelikož zdrojový článek [3] nespécifikuje jaký filtr použít, bylo v tomto projektu pro horní propust použito toto konvoluční jádro[27]:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Testováno bylo také toto jádro:

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Ale filtrování bylo zbytečně silné a výše uvedené tedy bylo označeno jako vhodnější.

- `void CrossBilateralFilter(pfstmo::Array2D *, pfstmo::Array2D *, pfstmo::Array2D *)` – Metoda velmi podobná na metodu `BilateralFilter`, pouze s menšími odlišnostmi. Nabízí se implementovat tyto dvě metody do jedné, a pouze volit požadovanou variantu parametrem. Pro přehlednost a jednoduchost bylo ale zvoleno implementovat tyto metody zvlášť. Podstatnou změnou oproti bilaterálnímu filtru je rozšíření velikosti filtrovacího okna, toto je třeba pro zachycení detailů. Experimenty bylo zjištěno, že vhodné je zvětšit toto okno dvojnásobně.
- `double CrossBilateralFilterWeight(double, double, double, double, pfstmo::Array2D *, pfstmo::Array2D *, bool)` – Obdoba `BilateralFilterWeight`, tentokrát však pro metodu `CrossBilateralFilter`.
- `int Transform()` – Hlavní funkce celého operátoru. Postupně jsou zde vytvořeny všechny potřebné histogramy, pole textureness, jsou aplikovány bilaterální filtry a také

se odsud volá histogram matching. Po naplnění všech dílčích polí se zde provádí výpočet finálního odstínu:

```
1 || shade = base(i, j) + rho(i, j) * detail(i, j);
```

Výpis 3.6: Závěrečný výpočet hodnoty jednoho bodu v metodě Bae06

Obrázek 3.11 ukazuje zlepšení kvality zvýšením počtu úrovní histogramu. Jak je vidět na obrázku 3.11b tak toto ještě nestačí pro dosažení dobrých výsledků při převodu HDR obrázků. Hlavní problém spočívá v tom, že v tomto operátoru se přenáší histogram base modelového obrázku na base konvertovaného obrázku. Pokud je ale vstupní obrázek HDR, je právě toto příčina efektu připomínajícího posterizace v převedených obrázcích, protože vysoký dynamický rozsah výstupního obrázku se zkrátka nemá jak „napasovat“ na omezený histogram modelu. To je tedy důvod, proč tato metoda poskytuje špatné výsledky při převodu HDR obrázků s nevhodně zvoleným modelem.

3.2 Tone mapping studio

Tone je mapping proces mapování HDR obrázků do prostoru zobrazitelného na současných monitorech, případně použitelného pro tisk, zkráceně tedy převod na LDR.

Tone mapping studio, označované zkratkou TMS, se stal základem pro implementaci metod pro převod barevných obrázků na černobílé. Název bývá někdy překládán jako „Techniky mapování tónů“. Jedná se o framework vyvíjený od roku 2004 pod vedením doc. Čadíka, který je pro implementaci takovýchto metod (operátorů) přímo určený. TMS tedy obsahuje rozhraní pro snadnou definici nových operátorů, včetně jejich typů. Dále se TMS stará o načítání a ukládání souborů. Nic nebrání použití TMS pro LDR obrázky, ale samozřejmě je zde podpora HDR formátu EXR a HDR. Přítomna je také podpora pro práci s některými barevnými modely.

Poznámka: Tone mapping studio byl naprogramován s využitím objektově orientovaného programování. V tomto kontextu má slovo metoda význam funkce náležící pod nějakou třídu. To je rozdílné oproti použití tohoto slova pro způsob převodu obrázků. Z důvodu, aby nedocházelo k nedorozuměním, bude v následujícím textu v případě situace týkající se převodu obrázků přednostně používáno slovo operátor. Slovo metoda pak bude „rezerováno“ pro použití v kontextu OOP.

Základní filozofie TMS je taková, že uživatel zvolí vstupní soubor, dále zvolí operátor pro aplikaci, nastaví jeho parametry a TMS tento operátor na vstupní soubor aplikuje a výsledek uloží do nového souboru. Nemusí se vždy jednat pouze o operátor vytvářející černobílý obrázek, může zde být implementován libovolný filtr. Převedené obrázky jsou ukládány současně do formátu TIFF a EXR.

Po přeložení vzniknou dva spustitelné soubory, pojmenované `tmocmd` a `tmogui`. Jedná se o variantu studia s grafickým rozhraní a bez něj. Tyto dva programy jsou základní funkcionalitou ekvivalentní. Aplikace `tmogui` nebyla při práci na tomto projektu použita.

Pro vytváření nových operátorů je důležitá třída `TMO`. Všechny operátory v rámci TMS dědí z této třídy, a nabízí se jim tedy předpřipravená paleta metod. Nejdůležitější z těchto metod je metoda `virtual int Transform()`, protože tato implementuje samotnou konverzi. Dále se zde nachází např. metody `virtual int SetName(const wchar_t*`

name) nebo `virtual int SetDescription(const wchar_t* description)` pro nastavení názvu a popisu daného operátoru. Kompletní seznam těchto metod se nachází v hlavičkovém souboru třídy TMO `src/TMS/tmolib/TMO.h`.

Druhou důležitou třídou je `TMOImage`. Tato třída zapouzdřuje všechny informace o souboru a poskytuje jakési rozhraní pro práci s obrázkem. Nejdůležitější metody z této třídy jsou: `virtual int Convert(int format, bool fast=false)` pro převod obrázku do reprezentace v jiném barevném modelu, `virtual int GetWidth()`, `virtual int GetHeight()` pro zjištění rozměrů obrázku, `inline double* GetPixel(int x, int y)` pro získání hodnot na konkrétním bodě v obrázku, a `virtual int ProgressBar(int part, int all)` pro snadné vypisování pokroku v aplikaci operátoru. Dále se zde nachází mimo jiné metody `virtual int Open(const char *filename)` pro otevření obrázku, `virtual int Save(int fileFormat=TMO_TIFF_8)` pro uložení obrázku, nebo `virtual int SaveWithSuffix(const char *suffix, int fileFormat=TMO_TIFF_8)` pro uložení obrázku s příponou v názvu. Tyto metody však již nejsou z pohledu uživatele frameworku tak důležité, protože uživatel je většinou sám neinvokuje. Jeden operátor může používat více instancí třídy `TMOImage`, pokud by pracoval s více vstupními soubory. Tohoto bylo využito při implementaci operátoru popsaného v 3.1.3.

TMS používá nástroj CMake⁴ pro generování Makefilů. Podrobnější informace o systému TMS lze nalézt v jeho dokumentaci, která se nachází v dokumentu `UI.pdf` umístěném na přiloženém DVD (A) ve složce `src/TMS/doc/`.

3.3 Implementace barevných modelů

Před začátkem práce na tomto projektu podporovalo TMS tyto barevné modely: RGB, XYZ, Yxy a Y (pouze jasová složka). Během práce na tomto projektu byla implementována podpora těchto barevných modelů: CIE $L^*a^*b^*$ (2.1.3), LCH (2.1.4) a CIE Luv (2.1.5). Tyto modely byly připsány do souboru `src/tmolib/TMOImage.cpp`.

Všechny tyto modely používají k reprezentaci jednoho bodu trojici proměnných s dvojnásobnou přesností (double). Všechny původní konverze byly umístěny v metodě `virtual int Convert(int format, bool fast=false)` a proto byly z důvodu dodržení konvence až na výjimky nové modely přidány také do této metody. Jednou z těchto výjimek je metoda `void TMOImage::LabToXyz(double L, double a, double b, double * x, double * y, double * z)`, která, jak již parametry naznačují, převádí hodnoty jediného bodu. K tomuto postupu bylo přistoupeno z důvodu potřeby převodu nějakého bodu, zatímco vstupní obrázek měl zůstat v původním barevném modelu. Toto bylo potřeba v metodě popsané v 3.1.1. Parametrem `format` metody `Convert` se specifikuje cílový barevný model.

Konkrétní matematické rovnice převodů byly popsány v kapitole 2.1. Z praktických důvodů nebyly implementovány vzájemné převody ze všech podporovaných modelů do všech. Namísto toho, pokud je požadavek na převedení např. z barevného modelu $L^*a^*b^*$ do Luv, provede se nejprve se převod z $L^*a^*b^*$ do XYZ a následně z XYZ do Luv. Přehled těchto postupů je zanesen do tabulky 3.1.

Symbol \times značí, že tento převod nebyl implementován, nebyl potřeba. Pro reprezentaci barevného modelu RGB byl použit barevný prostor sRGB. To je změna oproti barevnému prostoru, který byl v TMS původně implementován. Důvodem tohoto kroku bylo, že barevný prostor sRGB je více rozšířený, a snáze se tak ověřovala funkčnost. Pro převod z RGB do XYZ tedy byla použita tato matice:

⁴[//cmake.org](http://cmake.org)

do z	RGB	XYZ	L*a*b*	LCH	Luv
RGB	✓	RGB → XYZ	RGB → XYZ → L*a*b*	RGB → XYZ → L*a*b* → LCH	L*a*b* → XYZ → Luv
XYZ	XYZ → RGB	✓	×	×	XYZ → Luv
L*a*b*	L*a*b* → XYZ → RGB	L*a*b* → XYZ	✓	L*a*b* → LCH	L*a*b* → XYZ → Luv
LCH	LCH → L*a*b* → XYZ → RGB	×	LCH → L*a*b*	✓	×
Luv	×	×	×	×	✓

Tabulka 3.1: Přehled implementace převodů mezi barevnými modely.

$$M_{RGBtoXYZ} = \begin{bmatrix} 0.4124564 & 0.3575761 & 0.1804375 \\ 0.2126729 & 0.7151522 & 0.0721750 \\ 0.0193339 & 0.1191920 & 0.9503041 \end{bmatrix}$$

Illuminant byl použit D65. To znamená, že použité referenční hodnoty bílého bodu v modelu XYZ jsou (95,047; 100,0; 108,8830). Např. pro převod z L*a*b* do XYZ je ještě nutné použít konstanty κ a ϵ . V této implementaci byly použity novější standardy s hodnotami $\kappa = 24389/27$ a $\epsilon = 216/24389$. Všechny tyto hodnoty jsou uloženy ve formě maker v souboru `TMOImage.h`.

3.4 Použité knihovny

V této kapitole jsou velmi stručně představeny knihovny třetích stran, které byly použity v implementaci tohoto projektu.

3.4.1 `matrix.h`

Jak již název naznačuje, knihovna `matrix.h` [17] slouží pro práci s maticemi. Jejím autorem je Rondall E. Jones. Příjemné na této knihovně je, že je celá umístěna v jednom hlavičkovém souboru. Knihovna obsahuje třídy pro uložení vektoru, „řádku“, matice a diagonály. S těmito třídami je možné provádět klasické matematické operace. Kromě základních operací je zde také podpora pro maticové operace typu násobení, pseudoinverze, normalizace atd.

3.4.2 `poisson_pde`

Tato knihovna [19] zde slouží jako solver po Poissonovy rovnice (viz sekce 2.3.5). Algoritmus řešení je založen na rychlé Fourierově transformaci. Jedná se tedy o velmi rychlou metodu. Knihovna umožňuje specifikovat hraniční podmínky typu Neumann i Dirichlet.

V projektu se více osvědčil typ Dirichlet. Za nevýhodu této knihovny lze považovat její závislosti. První z nich je knihovna FFTW (3.4.5), druhou pak `boost::boost_multi_array` (3.4.4). Knihovna umí pracovat také s Laplaceovým operátorem 2.3.4 a je poměrně dobře dokumentovaná.

3.4.3 pfstools, pfstmo

Pfstools je sada nástrojů pro práce s HDR obrázky. Jednotlivé nástroje spolu komunikují pomocí rour. Části pfstools[22] byly v tomto projektu původně použity jako solver pro Poissonovy rovnice. Tato knihovna obsahuje mimo jiné metody `void solve_pde_multigrid (pfstmo::Array2D *F, pfstmo::Array2D *U)`, `void solve_pde_sor (pfstmo::Array2D *F, pfstmo::Array2D *U, int maxits=SOR_MAXITS)` a `void solve_pde_fft (pfstmo::Array2D *F, pfstmo::Array2D *U, bool adjust_bound=false)` pro různé algoritmy řešení. Později však byla pro tento účel použita knihovna `poisson_pde` (3.4.2), která se ukázala být snadněji použitelná. Pfstools se používá např. v programu Luminace HDR pro tvorbu HDR obrázků, který bude ještě zmíněn v kapitole 4.2.

Příbuzným k pfstools je knihovna pfstmo. Tato knihovna je zaměřená na tone mapping operátory. V projektu byl použit datový typ `pfstmo::Array2D` z této knihovny, který se dobře hodí pro ukládání obrázků v paměti počítače. Tento typ je definovaný v souboru `pfstmo.h`. Aby se usnadnila přenositelnost programu, byl tento soubor nakopírován do adresáře `TMS/tmolib`. Licence tohoto souboru takové zacházení umožňuje.

3.4.4 boost::multi_array

Na knihovně `boost::multi_array` [29] je závislá knihovna `poission_pde` (3.4.2). Boost je obecně velký balík C++ knihoven. Jednotlivé knihovny lze instalovat samostatně. `Multi_array` slouží pro práce s n-rozměrným polem. Typicky se používá např. pro vytvoření dvourozměrného pole, které bude mít několik „vrstev“ – vlastně tedy třetí rozměr.

3.4.5 FFTW

FFTW [15] slouží pro počítání diskrétní Fourierovy transformace. Tuto knihovnu používá knihovna `poission_pde` (3.4.2). Zajímavostí této knihovny je, že je možné ji volat z jazyka Fortran. FFTW podporuje vlákna, a je tedy možné ji efektivně využít na multiprocessorovém systému.

3.5 Použité vybavení

Program byl testován na počítači Asus UL80Jt. Tento stroj používá procesor Intel®Core i5 430UM a je osazen 4 GB operační pamětí DDR3 pracující na frekvenci 1066 MHz. Dále se v počítači nachází grafická karta NVIDIA®GeForce®310M a SSD pevný disk Kingston SSDNow V300.

Aplikace byla naprogramována v jazyce C++ s použitím knihoven popsanych v kapitole 3.4. Pro překlad byl použit překladač GCC ve verzi 4.9.2 a pro tvorbu Makefilů jsem používal nástroj cmake ve verzi 3.0.2. Použitým operačním systémem byl GNU/Linux Debian 3.16.

Fotografie pořízené za účelem vytvoření HDR testovacích snímků (4.1.2) byly fotografovány fotoaparátem Fujifilm FinePix S5800. To je fotoaparát z kategorie kompaktních. Tento fotoaparát je vybaven CCD snímačem o velikosti 1/2,5" a rozlišení 3264x2448, tedy přibližně 8 Mpx. Je schopný pracovat s rychlostí závěrky od 1/1000 s do 4 s. Clona je zde

použitelná v rozsahu od $f/3,5$ do $f/13,6$, nastavitelná po krocích $1/3$ EV. Zařízení podporuje ISO od hodnoty 64, přes 100, 200 atd. až po 1600 (fotografie s hodnotami ISO 800 a více jsou však již zatíženy značně viditelným šumem). Vestavěný objektiv snímá s ohniskovou vzdáleností od 6,33 mm po 63,3 mm. Velmi důležitou vlastností fotoaparátu podstatnou pro tuto práci je, že všechny tyto parametry lze nastavovat manuálně. Fotoaparát dokonce nabízí možnost manuálního ostření. Dále byl při fotografování použit stativ Tanner TR 301R.

Pro tvorbu grafů v této práci byl použit program MATLAB, jako editor zdrojových kódu byl používán grafický editor Kate, pro statistické výpočty posloužil webový nástroj Google Sheets. Dále byl využíván „GNU Image Manipulation Program“ pro různé úpravy obrázků a konzolová utilita **convert** pro hromadné zpracování obrázků.

Kapitola 4

Testování

V této kapitole je popsán způsob testování implementovaných metod. Metody byly testovány pomocí testovací sady (4.1), a také byl zjišťován subjektivní dojem v uživatelském experimentu pomocí dotazníku (4.3).

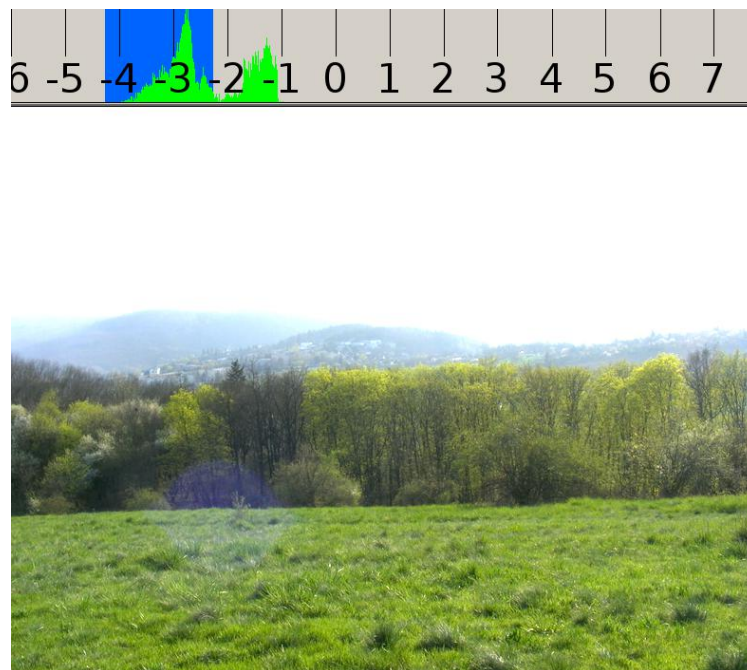
Tato kapitola a také některé jiné kapitoly obsahují HDR obrázky. Zobrazit HDR obrázky na papíře, tak aby byl zachován jejich dynamický rozsah nelze (viz. sekce 2.6.3). Jednou z možností, jak do této práce HDR obrázky umístit by bylo prostě transformovat celý jejich dynamický rozsah do LDR. Specifika této metody jsou popsány v 2.6.3. Zde však není použita, protože tím dochází k snížení kontrastu obrázku, který pak vypadá „mdle“. Aby byl zachován potenciál HDR, jsou všechny HDR obrázky v této práci zobrazeny spolu se svým histogramem a to tak, že není vždy vykreslen celý jejich dynamický rozsah, ale zobrazuje se pouze rozsah, který je na histogramu vyznačen modrou barvou. Toto je ukázáno na obrázku 4.1. V tomto konkrétním příkladě jsou vykresleny tmavší části obrázku (popředí), obloha je přepálená. Na histogramu je ale vidět, že obrázek obsahuje informace i o světlých místech, jen zde právě nejsou zobrazeny. Čísla v histogramu znamenají jednotlivé EV. Obrázek 4.1 tedy zobrazuje rozsah 2 EV. Do této práce byly obrázky vkládány s rozsahem nastaveným na „vhodný“, podle subjektivního názoru autora, nebo byl rozsah nastaven na nějakou zajímavou hodnotu. Na toto je třeba pamatovat při srovnávání HDR obrázků produkovaných různými metodami – jejich vyobrazení nezáleží jenom na převodové metodě, ale je také ovlivněno tím, jaká část histogramu byla použita.

Při vytváření HDR obrázků do tohoto textu bylo použito mapování gamma 2.2. Tím sice došlo k částečnému potlačení dynamického rozsahu, ale při použití lineárního mapování by bylo zprostředkování vysokého rozsahu ještě více omezené. Pro plnohodnotné prohlížení použitých HDR obrázků je doporučeno obrázky otevřít z příloženého DVD (A) ve vhodném prohlížeči.

U všech experimentů byly použity výchozí hodnoty všech parametrů. Pouze u metody Bae06 byl podle potřeby měněn modelový obrázek. U testovacích HDR obrázků cave, petrov a landscape byl použit model lowKey, u ostatních pak yosemite. U LDR obrázků byl vždy použit model yosemite.

4.1 Testovací sada

Pro testování metod pro převod barevných obrázků na černobíle byla použita testovací sada obrázků. Tuto sadu lze rozdělit na část LDR a HDR. Každá z těchto sad obsahuje 24 obrázků, celkem tedy bylo použito 48 testovacích obrázků. Všechny použité obrázky mají



Obrázek 4.1: Ukázka vyobrazení HDR obrázků v této práci. Nad obrázkem je umístěn jeho histogram, aktuálně používaná část histogramu je vyznačena modrou barvou.

poměrně nízké rozlišení. U LDR je to od 200×44 po 390×390 . Všechny HDR obrázky pak byly transformovány na rozlišení 800×600 , respektive 600×800 pro obrázky orientované „na výšku“. Toto nižší rozlišení však nemá žádnou zásadní nevýhodu, a naopak dovede při převezech velkého množství obrázků ušetřit čas, protože některé metody jsou poměrně časově náročné a urychlení je znatelné.

4.1.1 LDR

Testovací sada LDR obrázků byla převzata z [8]. Na straně 10 v této práci v popisu tabulky 5 je uveden odkaz na webovou stránku se zdrojovými obrázky. Úplný přehled těchto obrázků je zobrazen v příloze E. Zde se však z praktických důvodů nachází pouze výtah některých z nich, jejich kompletní přehled v plné velikosti lze nalézt na přiloženém DVD (A) ve složce `testSet`.

Tato sada pokrývá širokou škálu druhů obrázků. Jsou zde obyčejné fotografie (ovoce, květiny, portrét, motýl, makro), méně obvyklé obrázky (kresba, text), ale také např. vy-renderované obrázky. Podstatnou část sady tvoří také obrázky speciálně vytvořené pro testování metod `color2gray`.

4.1.2 HDR

Testovací HDR sada byla kompletně vytvořena v rámci této práce. Všechny fotografie byly vytvořeny klasicky s pomocí běžného digitálního fotoaparátu (bez vestavěné podpory HDR) spojováním několika snímků s různou expozicí do jednoho. Při vytváření této sady byla také snaha pokrýt různé fotografické žánry a fotografické styly.

Jak je již z podstaty HDR jasné, na většině z těchto snímků se ve větší či menší míře

vyskytuje protisvětlo. Nachází se zde velmi typické snímky interiéru s jasným oknem v záběru, a také záběry krajiny s jasnou oblohou, která by bez použití HDR byla přepálená. 87 % fotografií z této sady bylo pořízeno v exteriéru. Významnou část tvoří také noční fotografie, těch je asi 25 %.

Některé fotografie mají velmi vysoký dynamický rozsah (např. obrázek z jeskyně nebo pohled z místnosti na okno směřující ven), nachází se zde však hodně fotografií, které se nepokouší o extrémní dynamický rozsah. Důvodem toho je větší realističnost. Ve většině případů nemá scéna dynamický rozsah ve vysokých jednotkách EV, ale je jen mírně větší než dynamický rozsah snímáče.

I přes veškerou snahu některé fotografie obsahují „duchy“. To je způsobeno většinou větví, nebo obecně nějakou rostlinou v záběru, která se během fotografování hýbala ve větru. Může ale také jít o pohybující se dopravní prostředek. Takovéto předměty se s pomocí HDR techniky zachycují obtížně. Tyto fotografie byly v sadě ponechány, protože tím rozšiřují spektrum zkoumaných fotografií – v reálu se také může stát, že někdo zkoumané metody použije na takto poškozenou fotografii.

4.1.3 Modely

Kromě LDR a HDR obrázků obsahuje testovací sada také modely pro metodu Bae06 (3.1.3). Celkem byly použity 3 modely. Je mezi nimi jeden obrázek typu high-key (`highKey.tiff`), jeden obrázek typu lowKey (`lowKey.tiff`) a jeden „neutrální obrázek“ (`yosemite.tiff`). Tento obrázek byl převzat z [3] a vyznačuje se vysokým kontrastem. Obsah těchto obrázků dále není podstatný, protože slouží pouze pro přenos vzhledu. Obrázek `highKey.tiff` pochází z adresy¹ a obrázek `lowKey.tiff` z adresy².

4.1.4 Skript

Pro převod celé sady obrázků najednou byl použit skript `testAllMethods.sh`. Tento skript je uložen na příloženém DVD (A) ve složce `scripts`. `TestAllMethods.sh` má dva povinné parametry. Pomocí prvního z nich se určí, která složka bude zpracovávána, a druhým pak přípona souboru pro převod. Použití pro LDR obrázky ve formátu TIFF uložené ve složce LDR vypadá takto: `./testAllMethods LDR tiff` a použití pro HDR obrázky ve formátu exr uložené ve složce HDR by se provedlo takto: `./testAllMethods HDR exr`. Důležité je poznamenat předpoklad skriptu, že každý obrázek je dále umístěn v samostatné složce. Je to takto nastavené proto, že pokud by bylo více vstupních obrázků v jedné složce a bylo by použito více metod najednou, vzniklo by v této složce velké množství souborů a situace by byla nepřehledná. Na prvních řádcích skriptu jsou umístěny konfigurační proměnné, které upravují chování skriptu. Do proměnné `WORKING_FOLDER` je možné specifikovat, kde se nachází „základní“ pracovní adresář – v tomto adresáři se musí nacházet složka určená prvním parametrem. Proměnná `MODELS_FOLDER` určuje složku s modely pro metodu Bae06 (3.1.3). Vstupní obrázek bude touto metodou převeden pro všechny modely nacházející se v této složce. Skript také automaticky přejmenovává výstupní soubory z této metody, a to tak, že za jejich název přidává příponu podle použitého modelu. Díky tomu je možné převést jeden obrázek pomocí více modelů a nedochází k přepisování původních obrázků. Skript byl vytvořen čistě pro usnadnění práce při vytváření tohoto projektu, není tedy široce přenositelný a neošetřuje všechny možné chyby.

¹http://c1.staticflickr.com/5/4017/4279050501_5904e5a493_b.jpg

²http://serenephotographyblog.files.wordpress.com/2013/03/esha_lowkey-1-2.jpg

4.1.5 Struktura souborů

Tato sekce slouží k lepší orientaci v testovací sadě a obsahuje podrobnější informace k adresářové struktuře popsané v příloze A.

Testovací sada se ve složce `testSet`, která se nachází v kořenové adresáři DVD. Složka `testSet` obsahuje adresáře `LDR`, `HDR` a `models`, jak je naznačeno v odstavcích výše. Každý z adresářů `LDR` a `HDR` pak obsahuje 24 adresářů, každý pro jeden testovací obrázek. Každý z těchto adresářů obsahuje celkem 11 souborů s následujícím formátem názvu:

`<nazevObrazku>[<nazevMetody>[<_nazevModelu>]].<pripona>`

Pokud je obsažen pouze `<nazevObrazku>` a `<pripona>`, jedná se o originální vstupní barevný obrázek. Pokud je obsažen navíc `<nazevMetody>`, jedná se o obrázek převedený touto metodou, kde použitá metoda je buď `Kim09` nebo `Zheng15`. Pokud název obsahuje i `_nazevModelu`, jedná se o obrázek převedený metodou `Bae06` za použití tohoto modelu. Pro `LDR` obrázky byla použita `<pripona>` `<tiff>`, pro `HDR` obrázky pak `<exr>`. Formát TIFF byl zvolen pro svou bezztrátovou kompresi. Všechny převedené černobílé obrázky jsou uloženy s oběma těmito příponami. U `LDR` obrázků je uložení ve formátu `exr` zbytečné, bylo zde ponecháno pouze pro zachování jednotnosti.

Adresáře obrázků ve složce `HDR` obsahují kromě těchto 11 souborů navíc ještě adresáře `src` a `srcOriginal`. V adresáři `src` se nachází zdrojové fotografie použité pro vytvoření `.exr` `HDR` obrázku, je z nich možné vyčíst např. EXIF informace. Neplatí však, že by všechny fotografie ze složky `src` musely být skutečně použity – některé krajní expozice už mohly být pro použití do `HDR` obrázku nevhodné, nebo byl zvolený expoziční krok příliš jemný a použití všech obrázků by bylo zbytečné. Adresář `src` pak obsahuje nezmenšené obrázky přímo z fotoaparátu. Bohužel, použitý fotoaparát nepodporoval ukládání do formátu RAW, a tak jsou obrázky ve formátu JPEG.

4.2 Získávání HDR obrázků

HDR snímky pro tento projekt byly vytvořeny metodou popsanou v 2.6.1. Při fotografování byl fotoaparát nastaven na plně manuální režim. V prvním kroku byla nastavena citlivost na světlo. Při fotografování všech použitých snímků byla bez výjimky použita hodnota ISO 100. V druhém kroku se nastavovalo clonové číslo. Základní hodnotou tohoto parametru byla hodnota `f/3,5`. Pouze tam, kde byl snímek příliš světlý i pro nejkratší možný čas (`1/1000 s`) byla tato hodnota zvyšována a to až na hodnotu `f/13,5`. Tyto dvě hodnoty byly vždy stejné pro celou sérii snímků a měnil se expoziční čas – krok tři. Nejprve byla vytvořena nejtmaší fotografie s prokreslenými stíny s nejkratším časem, který se postupně prodlžoval až do vytvoření nejsvětější fotografie. Krok změny expozičního času byl různý a byl zvolen tak, aby celá série obsahovala přibližně 3 až 5 fotografií. Fotografováno bylo v maximálním možném rozlišení fotoaparátu, tedy `3264x2448`. Až na výjimky bylo vždy použito automatické ostření v režimu „kontinuální AF“ s režimem ostření „střed“. U žádné z fotografií nebyl použit blesk a všechny fotografie byly vytvořeny s použitím stativu. Parametry fotografie však bylo nutné nastavovat přímo na fotoaparátu, a proto i přes použití stativu došlo u některých fotografií k pohybu s fotoaparátem v rámci série.

Probíhaly také experimenty s funkcí „auto bracketing“ s nastavením ± 1 EV. S tímto nastavením sice prakticky nemůže dojít k pohybu fotoaparátu, ale také nelze přesně ovlivnit výslednou expozici. Z tohoto důvodu nebyly takto získané fotografie použity a všechny

použité fotografie byly získány čistě s manuálním nastavením pro získání plné kontroly nad vzhledem fotografie.

Po přenesení do počítače byly fotografie zmenšeny pro snadnější manipulaci. Spojování snímků do HDR fotografie probíhalo v softwaru Luminance HDR ve verzi 2.4.0 [2] (jedná se o multiplatformní open-source aplikaci pro práci s HDR obrázky vybavenou GUI).

Luminance HDR obsahuje průvodce pro tvorbu HDR obrázků. V prvním kroku se vyberou vstupní obrázky (aby program věděl, jak výsledný obrázek složit, musí u každého obrázku znát jeho expozici). Program Luminance HDR však pravděpodobně neumí automaticky stanovit EV z hodnot o expozici z EXIF informací u fotografie. Bylo tedy nutné hodnotu EV pro každou fotografii zadat ručně. Pro zjištění této hodnoty byl použit program `exif` z knihovny `libexif`. Dále se v prvním kroku průvodce nastavují možnosti automatického zarovnání fotografií a eliminace duchů. Ta v tomto projektu nebyla použita, nicméně automatické zarovnání bylo u sérií kde došlo k pohybu fotoaparátu použito a to v režimu „Hugin’s align_image_stack“. Toto automatické zarovnání trvá i u menších fotografií o pod jeden Mpx řádově jednotky minut, ale poskytuje poměrně dobré výsledky. Bylo také experimentováno s druhým režimem automatického zarovnání – MTB. Tato metoda však nefungovala správně a kvalitu fotografie naopak zhoršovala.

V druhém kroku průvodce se nastavují parametry algoritmu skládání fotografií. V základním režimu se nastavují parametry: váhová funkce, křivka odezvy a HDR model. Tyto parametry jsou seskupeny do šesti profilů, jejichž přehled je vypsán v následující tabulce 4.1.

	váhová funkce	křivka odezvy	HDR model
profil 1	trojúhelníková	lineární	Debevec
profil 2	trojúhelníková	gamma	Debevec
profil 3	plateau	lineární	Debevec
profil 4	plateau	gamma	Debevec
profil 5	Gaussova	lineární	Debevec
profil 6	Gaussova	gamma	Debevec

Tabulka 4.1: Tabulka nastavení programu Luminance HDR pro vytváření HDR obrázků

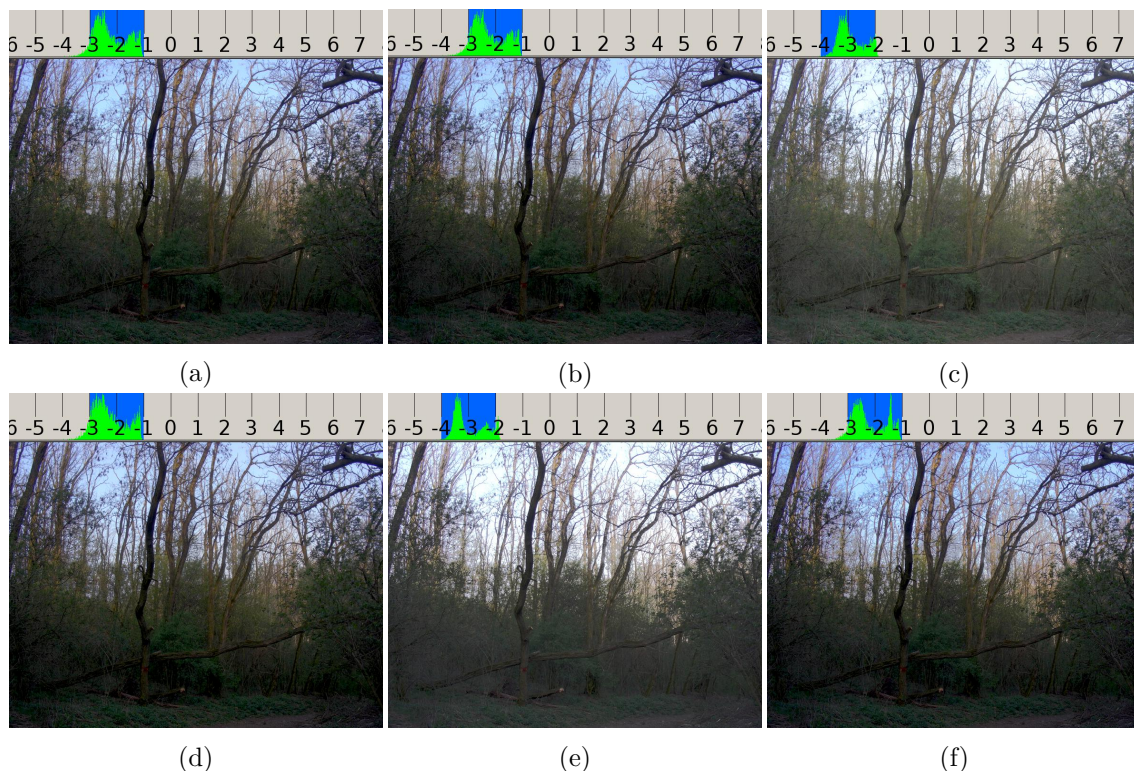
Porovnání obrázků vytvořených těmito profily se nachází v obrázku 4.2

Podstatné je také sledovat, jaký histogram daný profil vytvoří. Pro všechny obrázky vytvořené v rámci této práce byl použit profil 4, protože subjektivně poskytuje nejvyšší kvalitu výsledků. V Luminance HDR v kroku 2 při zvolení vlastní konfigurace je také možné upravit použitý HDR model a provést další nastavení. Tyto volby však nebyly v tomto projektu použity.

Hotové obrázky byly uloženy do formátu EXR. Po uložení ještě probíhalo ověření vzhledu obrázku pomocí programu `pfsview`. Ten umožňuje oproti Luminance HDR, např. po najetí myši na bod obrázku, zobrazit jeho polohu na histogramu.

4.3 Dotazník

Pro subjektivní porovnání kvality implementovaných metod byl použit dotazník. Z důvodu obtížného zobrazení HDR obrázků byla tato metoda použita pouze pro posouzení LDR obrázků. Pokud by dotazník měl obsahovat také HDR obrázky a byly by do něj tyto umístěny



Obrázek 4.2: Srovnání různých profilů pro tvorbu HDR obrázků v programu Luminance HDR. Obrázky 4.2a až 4.2f odpovídají profilům 1 až 6.

tak, jak jsou vyobrazeny v příloze E, mohlo by to vést k nepřesným výsledkům, protože odpovědi by byly závislé na tom, s jakým nastavením histogramu byl tento obrázek vytvořen. Forma dotazníku volně vychází z metody popsané v [8]. Specifické pro tento typ dotazníku je, že pro každý zkoumaný obrázek jsou položeny dvě otázky. Jedna otázka se zabývá přesností převodové metody, druhá pak preferencemi. Přesněji řečeno v první otázce (A) byly respondenti dotazováni na jejich osobní preference, tedy na černobílý obrázek, který se jim nejvíce líbí. Druhá otázka (B) je pak zaměřená na přesnost převodové metody. Respondenti měli zvolit, který z černobílých obrázků podle nich nejpřesněji reprezentuje originální obrázek. U obou těchto otázek viděli dotazovaní originální obrázek vedle převedených. To je také jedna z odlišností oproti [8], ve kterém se osobní preference vybírala v obou variantách, tedy bez referenčního obrázku i s ním.

Dotazník obsahoval 24 otázek (obrázků), tedy všechny z testovací LDR sady. Podobu jedné otázky ukazuje tabulka 4.2. Na konci dotazníku bylo ještě textové pole pro nepovinné zadání jména respondenta, jinak byly všechny otázky povinné.

V úvodu dotazníku byl odstavec s popisem úkolu. Jednotlivé otázky byly typu „radio button“. To znamená, že bylo možné (a také nutné) zvolit právě jednu odpověď. Po spuštění dotazníku nebyly ve výchozím stavu žádné odpovědi označeny a po volbě odpovědi ji ještě bylo možné libovolně změnit. Aby respondenti nebyli zatěžováni a ovlivňováni složitými názvy metod, byly odpovědi označeny pouze písmeny a, b, c.

Dotazník je minimálně do léta 2016 přístupný na adrese uvedené v poznámce³. Kdykoliv

³[//www.stud.fit.vutbr.cz/~xpospi68/DIP_dotaznik/](http://www.stud.fit.vutbr.cz/~xpospi68/DIP_dotaznik/)

Obrázek 4

A preference

Originál

☐ a

☐ b

☐ c

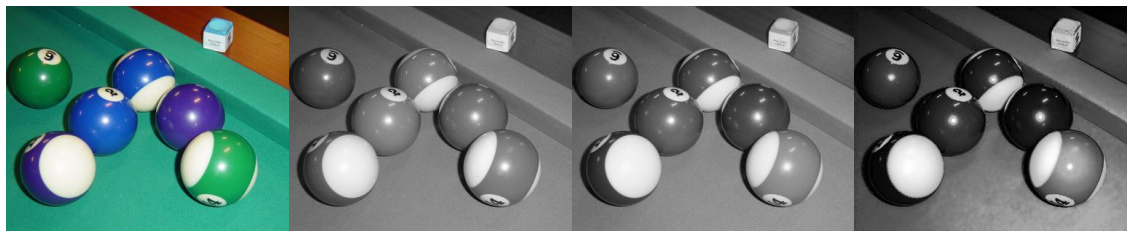
B přesnost

Originál

☐ a

☐ b

☐ c



Tabulka 4.2: Ukázka podoby jedné otázky z dotazníku.

potom jsou však jeho zdrojová data dostupná na přiloženém DVD **A** ve složce **questionnaire**. Data z dotazníku se ukládala do databáze na serveru `eva.fit.vutbr.cz`. Jedním z důvodů tohoto řešení je snadná analýza získaných dat pomocí jazyka SQL (agregační funkce, restrikce, atd.).

Výsledný vzhled obrázku převedeného metodou Bae06 je velmi výrazně závislý na volbě použitého modelu. Proto zde byla snaha o volbu „neutrálního“ modelu pro převod obrázků prezentovaných v dotazníku. Nakonec byl zvolen model yosemite **2.9b**, který byl použit u všech 24 obrázků. Důsledkem ale je, že u odpovědi na otázku B přesnost mohou být odpovědi ovlivněny tím, jak moc byl vzhled originálního obrázku podobný na vzhled modelového obrázku. Bylo by možné použít vstupní obrázek současně jako modelový obrázek a vznikl by černobílý obrázek se vzhledem podobným vstupnímu obrázku. Takový krok by ale byl plýtváním výpočetním výkonem, protože metoda Bae06 je navržena pro přenášení fotografického vzhledu a při takovéto operaci by probíhalo mnoho zbytečných operací.

4.3.1 Šíření dotazníku

Dotazník byl šířen s pomocí internetu. Snahou bylo, aby se dotazník dostal mezi co největší počet lidí, a aby si tito lidé byli co nejméně podobní, tedy aby odpovědi pocházely od širokého spektra obyvatelstva. Dotazování probíhalo třemi základními kanály. Prvním z nich bylo rozeslání mezi okruh známých autora. Druhým kanálem byly skupiny sociální sítě Facebook.

Žádost o vyplnění dotazníku byla sdílena do těchto skupin: VUT FIT MIT 2015-2017 Teambuilding⁴, VUT FIT 2BIT [2011-2014] GT⁵, 1. BIT 2012/2013 Teambuilding⁶, FAST VUT 2012-2016⁷, VUT Fakulta podnikatelská⁸, VUT FP IM 2014 - 2016⁹, PPV - Koleje

⁴ [//www.facebook.com/groups/930239423700911/](https://www.facebook.com/groups/930239423700911/)

⁵ [//www.facebook.com/groups/157039467702214/](https://www.facebook.com/groups/157039467702214/)

⁶ [//www.facebook.com/groups/406860272712986/](https://www.facebook.com/groups/406860272712986/)

⁷ [//www.facebook.com/groups/442831409166748/](https://www.facebook.com/groups/442831409166748/)

⁸ [//www.facebook.com/groups/34928898905/](https://www.facebook.com/groups/34928898905/)

⁹ [//www.facebook.com/groups/336030009888114/](https://www.facebook.com/groups/336030009888114/)

pod Palackého vrchem (VUT Brno)¹⁰, Koleje VUT (Palačák)¹¹, Purkyňovy koleje (VUT Brno)¹², Brněnská PAVLAČ a SEZNAMKA u orloje¹³, Tauferovy koleje Brno¹⁴, FEKT VUT BRNO - KEL2¹⁵, Brno's vegans¹⁶, Koleje Kounicova¹⁷, VŠB-TUO FEI Mgr. 2014-2016 Informatika a výpočetní technika¹⁸, MENDELU FRMS 2013¹⁹, Filozofická fakulta MU Brno²⁰.

Posledním použitým kanálem byla internetová fóra. Dotazník byl vložen do těchto: VUTRoom²¹, Fituška²².

Účelem dotazníku bylo posoudit vlastnosti implementovaných metod bez jakýchkoli dalších souvislostí. Cílem nebylo zjistit nějakou souvislost mezi odpovědí a skupinou respondentů. Proto v dotazníku nebyly obsaženy např. otázky typu věk, nebo pohlaví.

¹⁰ [//www.facebook.com/groups/393802530659892/](https://www.facebook.com/groups/393802530659892/)

¹¹ [//www.facebook.com/groups/205362359560932/](https://www.facebook.com/groups/205362359560932/)

¹² [//www.facebook.com/groups/pkolejevutbr/](https://www.facebook.com/groups/pkolejevutbr/)

¹³ [//www.facebook.com/groups/801764056558410/](https://www.facebook.com/groups/801764056558410/)

¹⁴ [//www.facebook.com/groups/Tauferky/](https://www.facebook.com/groups/Tauferky/)

¹⁵ [//www.facebook.com/groups/470605629636642/](https://www.facebook.com/groups/470605629636642/)

¹⁶ [//www.facebook.com/groups/404769409571494/](https://www.facebook.com/groups/404769409571494/)

¹⁷ [//www.facebook.com/groups/168879651857/](https://www.facebook.com/groups/168879651857/)

¹⁸ [//www.facebook.com/groups/vsb.fei.bc.2011.2014.ivt/](https://www.facebook.com/groups/vsb.fei.bc.2011.2014.ivt/)

¹⁹ [//www.facebook.com/groups/656337107727953/](https://www.facebook.com/groups/656337107727953/)

²⁰ [//www.facebook.com/groups/44452741109/](https://www.facebook.com/groups/44452741109/)

²¹ [//student.vutbr.cz/forum/](https://student.vutbr.cz/forum/)

²² [//fituska.eu/](https://fituska.eu/)

Kapitola 5

Výsledky

Tato kapitola obsahuje srovnání výsledků implementovaných metod na testovací sadě obrázků.

5.1 Posouzení převedených obrázků

V této práci není prostor pro hodnocení všech 48 otestovaných obrázků, proto bude vybráno a posouzeno pouze několik nejzajímavějších. Celá galerie testovacích obrázků se nachází v příloze E. Aby nedocházelo k nejasnostem, budou obrázky vždy umístěny v tomto pořadí: zleva originální obrázek, výstup metody Kim09, výstup metody Zheng15, výstup metody Bae06.

5.1.1 LDR

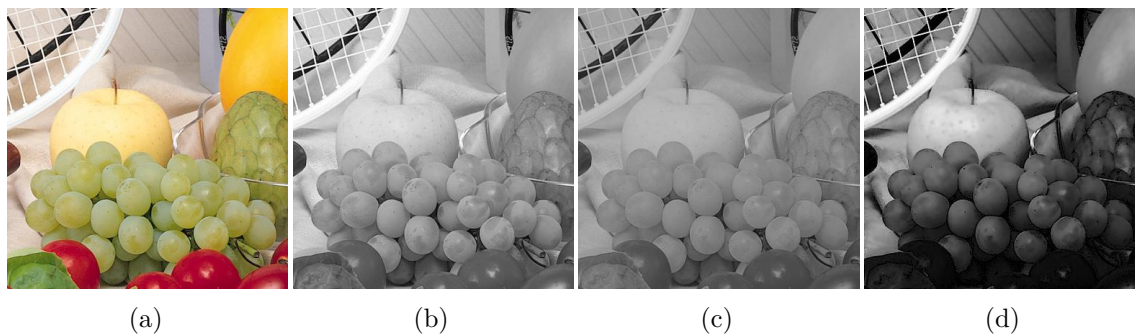
Tato sekce bude oproti 5.1.2 stručnější, protože vyhodnocení obrázků se věnuje samostatná podkapitola o vyhodnocení dotazníku. (5.2).

Jak již bylo naznačeno v kapitole 4.3, obrázky z metody Bae06 není vhodné přímo srovnávat s obrázky z ostatních dvou metod z hlediska např. přesnosti, protože tato metoda spadá do trochu jiné kategorie. Výstupy z metody Bae06 velmi silně závisí na na použitém modelu.

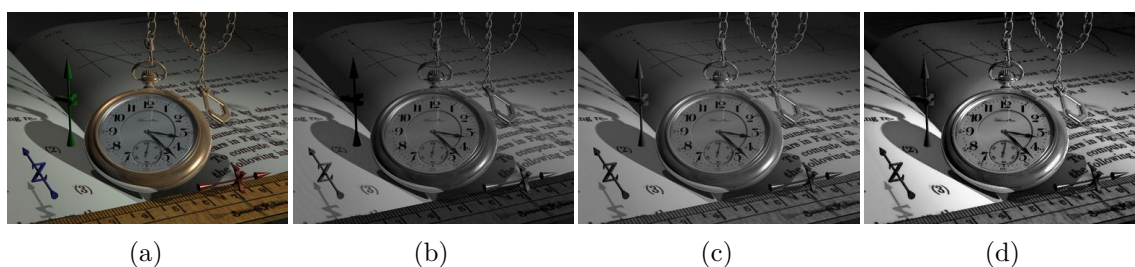
Obecně lze říci, že metoda Kim09 pracuje skutečně robustně, protože jí převedené obrázky nevykazují žádné na první pohled viditelné chyby způsobené převodem. Navíc jsou obrázky z této metody oproti Zheng15 o něco více kontrastní. Pro konkrétní obrázek 5.1 má metoda Kim09 oproti Zheng15 také více detailů. To je viditelné např. na povrchu jablka. Tento jev by bylo možné ještě posílit navýšením parametru *alpha*. Obrázek Zheng15 zde působí plošeji, ne tak výrazně. Obrázek z metody Bae06 vykazuje velmi vysoký kontrast, to je způsobeno použitím modelu „yosemite“.

Naopak na dalším testovacím obrázku 5.2 je kontrastnější obrázek z Zheng15. Díky tomu je také text v pozadí lépe čitelný. Obrázek Zheng15 zde vypadá také přirozeněji, protože např. pravitko v popředí není v originální obrázku tak tmavé, jak jej prezentuje obrázek Kim09.

Obecně produkuje metoda Zheng15 o něco světlejší obrázky než Kim09. Toto je vidět např. na obrázcích 15 nebo 16 dostupných v příloze E. Metoda Zheng15 se občas chová skoro jako by pracovala s ekvalizací histogramu – takový rozsah produkované obrázky mají. Metoda Kim09 ale díky tomu produkuje věrnější výsledky.



Obrázek 5.1: Srovnání výstupů pro vstupní obrázek 10 – fruits. 5.1a – vstup, 5.1b – Kim09, 5.1c – Zheng15, 5.1d – Bae06



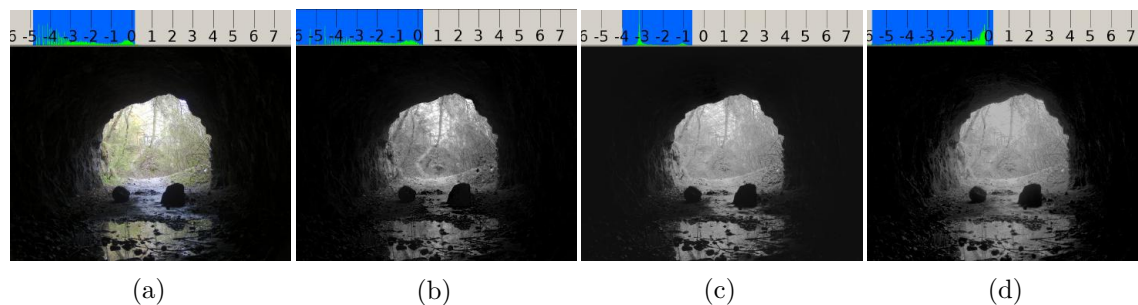
Obrázek 5.2: Srovnání výstupů pro vstupní obrázek 24 – watch. 5.2a – vstup, 5.2b – Kim09, 5.2c – Zheng15, 5.2d – Bae06

5.1.2 HDR

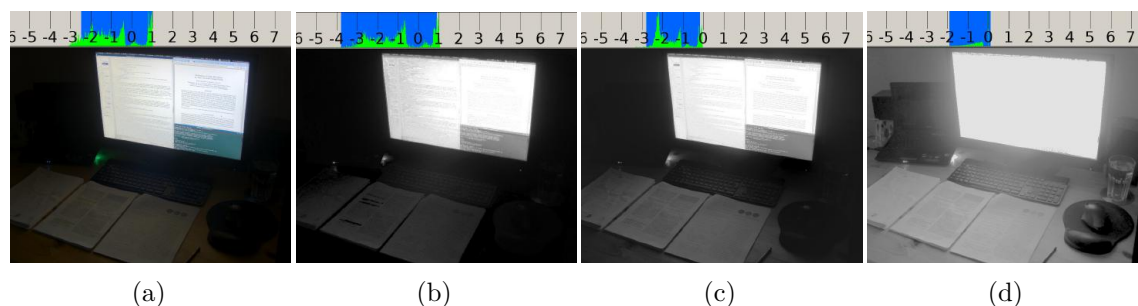
Ve srovnání s pomocí HDR obrázků se dají mezi metodami očekávat větší rozdíly, protože na toto tyto metody nejsou stavěné.

Jak již bylo naznačeno v kapitole 3.1.3, největší problémy zde má metoda Bae06. Při použití LDR modelu je to způsobeno tím, že metoda Bae06 se pokouší přenést histogram modelového obrázku na HDR histogram vstupního obrázku. Modelový obrázek ale většinou obsahuje buď žádné, nebo je jen minimum odstínů na těch intenzitách jasu, na kterých jich má nejvíc vstupní obrázek. Z toho důvodu pak není odstíny vstupního obrázku na co přenášet a na výstupu se to projevuje efektem posterizace nebo rozmazání. Tuto situaci by bylo možné řešit implementací speciálního, nejlépe 32-bitového, HDR histogramu do metody Bae06. Cílem této práce ale bylo porovnat chování metod tak, jak jsou popsány v článcích, a proto tento krok nebyl realizován. Navíc zbývající metody Kim09 a Zheng15 také nebyly žádným způsobem pro zpracování HDR obrázků upravovány, takže vylepšení metody Bae06 by mohlo ohrozit nestrannost srovnávání těchto metod. I přesto zde k jistým drobným optimalizacím pro HDR obrázky došlo, např. implementace „posilovače“ histogramu (HISTOGRAM_BOOSTER). Toto však sloužilo pouze k potvrzení teorie o tom, kde je problém. Opravdový HDR histogram by musel být implementován v plovoucí řádové čárce. Implementace histogram matchingu v plovoucí řádové čárce by pak s sebou mohla přinést komplikace, protože takový proces by byl výpočetně velmi náročný. Na druhou stranu např. u obrázku landscape má metoda Bae06 velmi dobrý výsledek. To je částečně způsobeno použitím modelu lowKey.

Obě metody Kim09 a Zheng15 převádějí HDR obrázky na černobílé úspěšně. Velkým plusem metody Kim09 je, že ve většině případů velmi přesně zachovává tvar histogramu vstupního obrázku. Přesto se v této disciplíně ukázala jako nejrobustnější metoda Zheng15, protože, ačkoliv ve většině pokusů funguje Kim09 uspokojivě, v některých extrémních situacích dochází k zanesení šumu do obrázku. V některých případech došlo po aplikaci metody Kim09 k zviditelnění jinak nepatrných artefaktů způsobených uložením vstupního obrázku s JPEG kompresí. Problém metody Zheng15 je, že mírně snižuje dynamický rozsah obrázků. Toto bylo potvrzeno programem `exrtools` [4].

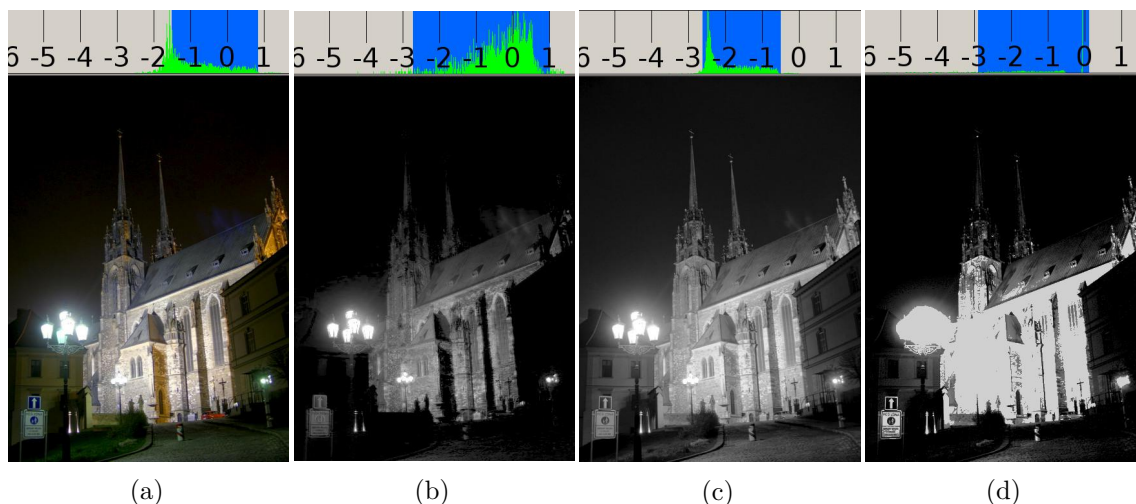


Obrázek 5.3: Porovnání výstupů metod pro vstupní obrázek „cave“. Na tomto obrázku je vidět, jak Kim09 (5.3b) velmi přesně zachoval histogram vstupního obrázku. U tohoto vzorku pravděpodobně dopadla metoda Kim09 dokonce lépe než Zheng15 (5.4c), u kterého je histogram komprimovaný. Také metoda Bae06 (5.3d) zde poskytuje překvapivě dobré výsledky. Je to dáno hlavně volbou vhodného modelu, v tomto případě lowKey, který přibližně odpovídá vstupnímu obrázku, a proto byl převod relativně úspěšný.



Obrázek 5.4: Porovnání výstupů metod pro vstupní obrázek „monitor“. Nijak neporušený obraz produkuje metoda Zheng15 (5.4c), i když histogram je opět částečně omezen. U metody Kim09 (5.4b) je jasně viditelný obrazový šum a také je zde pozorovatelné mírné rozostření na přechodu tmavá - jasná (hrana monitoru). U metody Bae06 (5.4d) je vidět dopad přenesení histogramu na histogram modelu – informace o obrazu na monitoru se ztratily. Na podložce od myši je vidět silný šum; toto však nemusí být chyba samotné metody, tento typ šumu je poměrně často možné omezit použitím většího okna u bilaterálního filtru (pak ale výpočet trvá velmi dlouho).

Z popsanych obrázků by se mohlo zdát, že se metoda Kim09 pro HDR nehodí. Uvedené obrázky však prezentují spíše extrémní situace, přibližně u 75 % z testovací HDR sady poskytovala metoda Kim09 uspokojivé výsledky. Největší problémy má metoda Kim09 s noč-



Obrázek 5.5: Porovnání výstupů metod pro vstupní obrázek „petrov“. Tento obrázek sem byl vybrán proto, že ukazuje, jak špatně může výstup jinak poměrně kvalitní metody Kim09 (5.5b) dopadnout. Celý obrázek je velmi silně zašuměný a např. v oblasti nad lampou se objevily artefakty typické pro JPEG kompresi. Metoda Zheng15 (5.5c) podává dobrý výsledek. U metody Bae06 (5.5d) dochází k silným přepalům a histogram je zde velmi plochý. Použit byl model lowKey.

ními fotografiemi, kde v jinak tmavém obrázku září jasný zdroj světla. Metoda Zheng15 neselhala v žádném z obrázků z testovací HDR sady, a lze ji tedy označit jako vhodnou pro použití s HDR obrázky.

5.2 Data z dotazníku

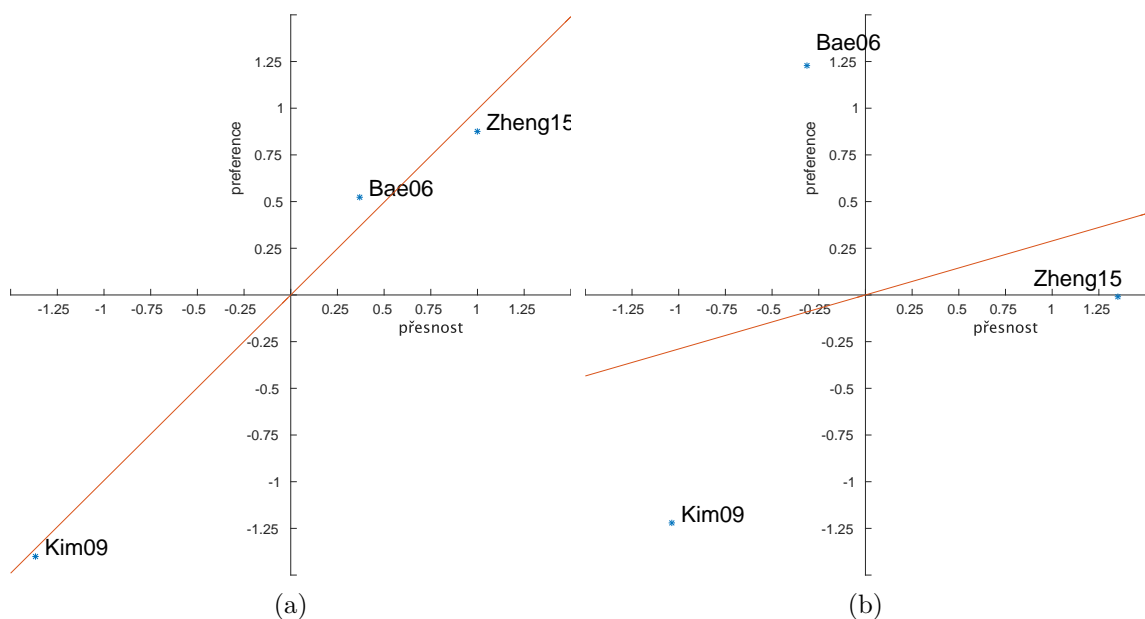
Díky efektivnímu šíření dotazníku, popsanému v kapitole 4.3.1 bylo celkem obdrženo 540 vyplněných dotazníků. To je poměrně vysoké číslo, data by tedy měla být validní. Kompletní surová data z dotazníku se nachází na příloženém DVD A ve složce questionnaire v souboru allData.sql ve formátu SQL. Po importování tohoto souboru do MySQL databáze lze zjistit např. počty hlasů pro jednotlivé odpovědi. Pro obrázek 1 se to provede tímto dotazem:

```
SELECT a1, COUNT( a1 ) FROM dipQuestionnaireAnswers GROUP BY a1 UNION ALL
SELECT p1, COUNT( p1 ) FROM dipQuestionnaireAnswers GROUP BY p1.
```

V práci [8] se pro výpočet standardizovaného skóre používá tzv. „zákon porovnávání soudů“. To zde nelze použít, protože zatímco v [8] vybírali respondenti pouze ze dvou možností, v tomto dotazníku bylo na výběr ze všech testovaných metod. Jako standardní skóre bylo nakonec zvoleno nejjednodušší z-skóre s výpočtem podle rovnice [13]:

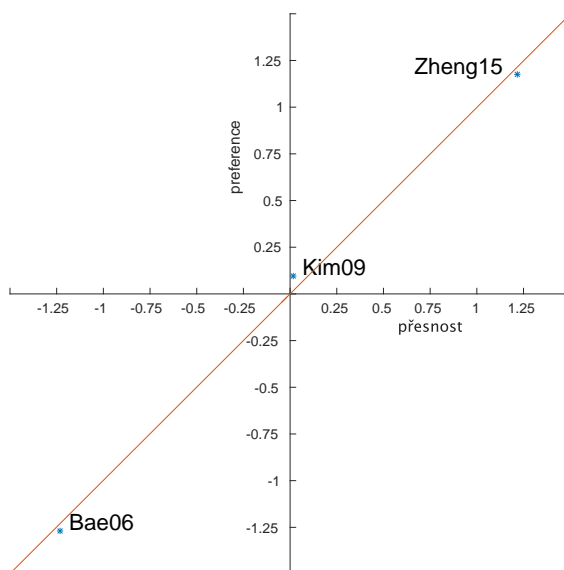
$$z = \frac{X - \mu}{\sigma} \quad (5.1)$$

V rovnici 5.1 představuje X původní hrubé skóre, μ je původní průměrná hodnota, σ je původní směrodatná odchylka a z je získané z-skóre.



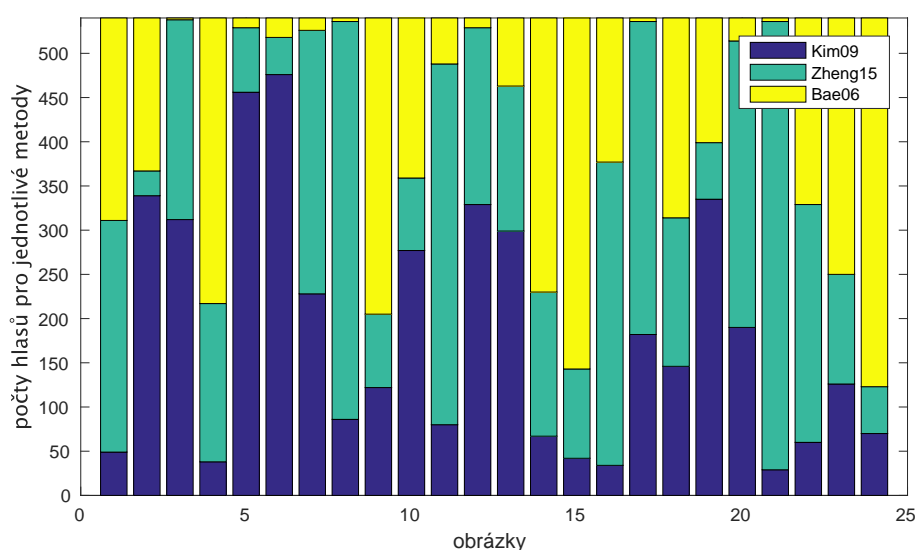
Obrázek 5.6: Grafické zobrazení vztahu přesnosti a preferencí pro obrázky 1 (5.6a) a 4 (5.6b).

Na obrázku 5.6 se nachází vybrané grafy vztahu přesnosti a preferencí. Kompletní přehled pro všechny obrázky z testovací sady je umístěn v příloze E.

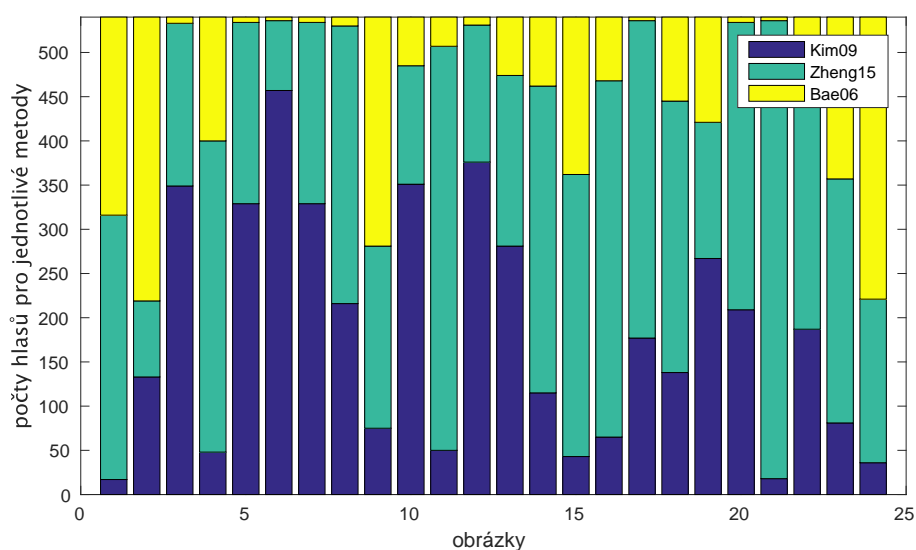


Obrázek 5.7: Celkové výsledky dotazníku

Ve většině případu dopadly výsledky „podle očekávání“. Tím je myšleno, že některý obrázek dopadl nejlépe, jiný nejhůře a poslední byl někde uprostřed a to jak v dimenzi přesnosti, tak preferencí. Toto je vidět na obrázku 5.6a. Občas ale nastala i taková situace, že bod byl umístěn do 2. kvadrantu. To znamená, že respondenti často označovali takovýto



Obrázek 5.8: Celkové preference



Obrázek 5.9: Celková přesnost

obrázek jako preferovaný, ale neoznačovali jej jako přesný. Toto ukazuje obrázek 5.6b. Ještě výraznější „anomálie“ lze pozorovat u obrázků 14 a 15. U těchto obrázků dominovala v preferencích metoda Bae06. S velkou pravděpodobností je to způsobeno tím, že tyto obrázky mají velmi vysoký kontrast způsobený použitím modelu yosemite. Grafy těchto situací je možné nalézt v příloze E. U obrázku 23 došlo k opačné situaci. Metoda Zheng15 se umístila ve 4 kvadrantu. To znamená, že byla hodnocena jako velmi přesná, přesto však málo preferovaná. Důvodem nízkých preferencí je pravděpodobně vyšší kontrast u ostatních metod. Z grafů je vidět, že lidé nejvíce preferují obrázky s vysokým kontrastem. Pokud má některý

z převedených obrázků výrazně větší kontrast než ostatní, velmi často je také tím, který má nejvíce preferencí.

Obrázek 5.7 ukazuje celkové výsledky z dat nasbíraných na celé testovací sadě. Na první pohled je patrné, že korelace přesnosti a preference je velmi vysoká. Celkově dopadla podle tohoto dotazníku nejlépe metoda Zheng15; má nejvyšší přesnost i preference. Nejhůře dopadla Bae06, to je ale dáno tím, že jsou uvažovány hodnoty ze všech otázek, u konkrétních otázek tato metoda často v preferencích dominuje. Mezi těmito se umístila metoda Kim09.

Graf 5.8 zachycuje rozložení preferencí u jednotlivých otázek. Graf 5.9 pak znázorňuje totéž pro přesnost. Korelace mezi jednotlivými obrázky je nízká. Je vidět, že u některých obrázků metoda Bae06 silně dominuje v preferencích (tam, kde ostatní metody nemají takový kontrast – obrázky 4, 9, 14, 15 a 24), tam, kde ale vznikly nějaké chyby v obraze, tato metoda pochopitelně velmi silně ztrácí (obrázky 5, 6, 7, 12, 17, 20 a 21). Celkové preference mezi metodami Kim09 a Zheng15 jsou velmi vyrovnané, i když těsně vyhrává Zheng15. Preference se velmi liší podle obrázků.

V případě přesnosti také vyhrává metoda Zheng15, získala téměř 50 % všech odpovědí, hlavně na úkor metody Bae06. Metoda Kim09 vyhrála zhruba 35 % ze všech odpovědí na otázku přesnosti, což lze považovat za dobrý výsledek. Nejnižší celkovou přesnost má metoda Bae06 s podílem jen málo přes 15 %. Tyto výsledky jsou však očekávané, protože metoda Bae06 nemá za cíl vysokou přesnost (výsledek je ovlivněn modelem).

Kapitola 6

Závěr

Cílem této práce bylo implementovat metody pro převod barevných obrázků na černobílé a posoudit jejich vlastnosti. Tento záměr byl splněn.

V rámci řešení diplomové práce byly nejprve vybrány a popsány tři metody vhodné pro implementaci. Všechny zvolené metody jsou použitelné z fotografického hlediska. Následně byly tyto metody také implementovány, a to jako komponenty do již existujícího systému TMS. V dalším kroku byla vytvořena testovací sada 24 HDR obrázků. Při jejich tvorbě byl použit běžný fotoaparát; snímky z něj byly s pomocí speciálního software složeny do HDR fotografií. Testovací sada LDR obrázků byla převzata od vedoucího práce. S pomocí těchto testovacích sad bylo s metodami experimentováno, a byly posuzovány jejich vlastnosti z hlediska převodu HDR obrázků. Závěry z těchto experimentů byly popsány, bylo také provedeno vyhodnocení kvality převodu LDR obrázků s pomocí dotazníku. Nakonec byl vytvořen plakát shrnující výsledky této práce.

Experimenty bylo zjištěno, že při převodu HDR obrázků poskytuje velmi dobré výsledky metoda Zheng15. Jediným jejím neduhem je, že v některých případech mírně zmenšuje dynamický rozsah obrázku. Dobré výsledky produkuje také metoda Kim09. Pouze u malého procenta testovaných obrázků došlo k výraznějšímu zhoršení obrazové kvality. Nejhorší v této disciplíně dopadla metoda Bae06, která se v implementaci s obyčejným histogramem pro převod HDR obrázků nehodí. Metody byly vyhodnoceny v uživatelském experimentu, kterého se zúčastnilo 540 osob.

Práce mne zasvětila do problematiky převodu barevných obrázků na černobílé. Kromě hlubšího pochopení, jak tyto metody fungují, jsem také získal zkušenosti s prací s HDR obrázky. Jejich vytváření a experimenty s nimi mne bavily a věřím, že tato technologie je pouze na počátku svého širokého rozšíření.

V práci by bylo možné pokračovat rozšířením metody Bae06 o podporu HDR histogramu, aby i ona přinášela dobré výsledky při převádění HDR obrázků. Další možností pokračování v práci by bylo implementování zcela nové metody pro převod barevných obrázků na černobílé, která by byla od začátku zaměřena na HDR. Jsou to právě černobílé obrázky, které na HDR zobrazovačích velmi dobře vyniknou, protože pozorovatele nebude od dojmu z vysokého dynamického rozsahu nic rozptylovat. Toto téma je velmi aktuální, protože HDR displeje právě přicházejí na trh.

Literatura

- [1] Adi Ben-Israel, T. N. G.: *Generalized Inverses*. 640 Bartholomew Rd, Piscataway, NJ 08854-8003, USA: Rutcor-Rutgers Center for Operations Research, Rutgers University, 2001.
- [2] Anastasia, D.: Luminance HDR. 2015 [cit. 2016-05-10].
URL <http://qtpfsgui.sourceforge.net/>
- [3] Bae, S.; Paris, S.; Durand, F.: Two-scale Tone Management for Photographic Look. *ACM Trans. Graph.*, ročník 25, č. 3, Červenec 2006: s. 637–645, ISSN 0730-0301, doi:10.1145/1141911.1141935.
URL <http://doi.acm.org/10.1145/1141911.1141935>
- [4] Biggs, B.: exrtools. 2004 [cit. 2016-05-05].
URL <http://scanline.ca/exrtools/>
- [5] Blinn, J.: *Jim Blinn's Corner: Dirty Pixels (The Morgan Kaufmann Series in Computer Graphics)*. Morgan Kaufmann, 1998, ISBN 1558604553.
- [6] Boitard, R.; Mantiuk, R. K.; Pouli, T.: Evaluation of color encodings for high dynamic range pixels. *Proc. SPIE*, ročník 9394, 2015: s. 93941K–93941K–9, doi:10.1117/12.2077715.
URL <http://dx.doi.org/10.1117/12.2077715>
- [7] Bočík, A.: *Velká kniha HDR fotografie*. Computer Press, 2011, ISBN 978-80-251-3367-5.
- [8] Čadík, M.: Perceptual Evaluation of Color-to-Grayscale Image Conversions. *Comput. Graph. Forum*, ročník 27, č. 7, 2008: s. 1745–1754.
- [9] Cooper, T.: *The Realistic HDR Image (Fuel)*. Peachpit Press, 2014, ISBN 978-0-13-384564-8.
- [10] Demmel, J.: CS267: Notes for Lectures 15 and 16, Mar 5 and 7, 1996. 1996 [cit. 2016-03-20].
URL <http://www.cs.berkeley.edu/~demmel/cs267/lecture24/lecture24.html>
- [11] Farbman, Z.; Fattal, R.; Lischinski, D.; aj.: Edge-Preserving Decompositions for Multi-Scale Tone and Detail Manipulation. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2008)*, ročník 27, č. 3, Srpen 2008.
- [12] Fisher, R.; Perkins, S.; Walker, A.; aj.: Gaussian Smoothing. 2003 [cit. 2016-04-15].
URL <http://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm>

- [13] Foley, H. J.: Chapter 5: z-Scores.
URL <http://www.skidmore.edu/~hfoley/Handouts/S.Ch5.pdf>
- [14] Freeman, M.: *Základy HDR: fotografie a vysoký dynamický rozsah*. Brno: Zoner Press, 2008, ISBN 978-80-86815-95-4.
- [15] Frigo, M.; Johnson, S. G.: The Design and Implementation of FFTW3. *Proceedings of the IEEE*, ročník 93, č. 2, 2005: s. 216–231, special issue on “Program Generation, Optimization, and Platform Adaptation”.
- [16] Hrivňák, D.: *Diferenciální operátory vektorové analýzy*. Ostravská univerzita přírodovědecká fakulta, 2002.
URL http://artemis.osu.cz/uvma3/UVMA3_1.pdf
- [17] Jones, R.: Solutions for Systems of Linear Equations. 2010 [cit. 2015-11-03].
URL <http://www.rejones7.net/rejtrix/>
- [18] Kim, Y.; Jang, C.; Demouth, J.; aj.: Robust Color-to-gray via Nonlinear Global Mapping. *ACM Transactions on Graphics (SIGGRAPH ASIA 2009)*, ročník 28, č. 5, Prosinec 2009.
- [19] Kluge, T.: Fast direct solver for the 2D-Poisson’s equation in C++. 2008 [cit. 2015-11-20].
URL http://kluge.in-chemnitz.de/opensource/poisson_pde/
- [20] Kovár, M.: *Maticový a tenzorový počet*. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav matematiky, 2005.
- [21] Lindbloom, B. J.: Interesting things for digital imaging and color science. 2014 [cit. 2016-01-02].
URL <http://www.brucelindbloom.com/>
- [22] Mantiuk, R.: pfstools About. 2015 [cit. 2015-12-02].
URL <http://pfstools.sourceforge.net/index.html>
- [23] Murray, J. D.; vanRyper, W.: *Encyklopedie grafických formátů*. Computer Press, 1997, ISBN 80-7226-033-2.
- [24] Nayatani, Y.: Simple estimation methods for the Helmholtz-Kohlrausch effect. *Color Research & Application*, ročník 22, č. 6, Prosinec 1997: s. 385–401.
- [25] Paris, S.; Kornprobst, P.; Tumblin, J.; aj.: Bilateral Filtering: Theory and Applications. *Foundations and Trends® in Computer Graphics and Vision*, ročník 4, č. 1, 2008.
URL http://people.csail.mit.edu/sparis/publi/2009/fntcgv/Paris_09_Bilateral_filtering.pdf
- [26] Pavel Drábek, G. H.: *Parciální diferenciální rovnice*. Západočeská univerzita v plzni, 2011.
URL http://mi21.vsb.cz/sites/mi21.vsb.cz/files/unit/parcialni_diferencialni_rovnice.pdf
- [27] Phillips, D.: *Image Processing in C*. Cmp Books, 1994, ISBN 0131045482.

- [28] Reinhard, E.; Heidrich, W.; Debevec, P.; aj.: *High Dynamic Range Imaging, Second Edition: Acquisition, Display, and Image-Based Lighting*. Morgan Kaufmann, 2010, ISBN 012374914X.
- [29] Ronald Garcia, A. L., Jeremy Siek: Boost.MultiArray. 200-2001 [cit. 2015-11-04].
URL http://www.boost.org/doc/libs/1_56_0/libs/multi_array/doc/index.html
- [30] Szeliski, R.: *Computer Vision: Algorithms and Applications*. New York, NY, USA: Springer-Verlag New York, Inc., první vydání, 2010, ISBN 1848829345, 9781848829343.
- [31] Young, T.; Mohlenkamp, M. J.: *Introduction to Numerical Methods and MATLAB Programming for Engineers*. Department of Mathematics, Ohio University, Athens, OH 45701, 2015.
URL <https://www.math.ohiou.edu/courses/math3600/book.pdf>
- [32] Zheng, X.; Feng, J.; Zhou, B.: Efficient Color-to-gray Conversion for Digital Images in Gradient Domain. In *Proceedings of the 14th ACM SIGGRAPH International Conference on Virtual Reality Continuum and Its Applications in Industry, VRCAI '15*, New York, NY, USA: ACM, 2015, ISBN 978-1-4503-3940-7, s. 85–88, doi:10.1145/2817675.2817687.
URL <http://doi.acm.org/10.1145/2817675.2817687>
- [33] Zhinan, X.: Optimization-based Image Decolorization. 2012.
URL <http://web.cs.ucla.edu/~zhinan/up/Optimization-based%20Image%20Decolorization.pdf>
- [34] Žára, J.; Beneš, B.; Sochor, J.; aj.: *Moderní počítačová grafika*. Computer Press, 2004, ISBN 80-251-0454-0.

Přílohy

Seznam příloh

A Obsah DVD	74
B Plakát	75
C Překlad a instalace programu	76
C.0.1 Závislosti	76
C.0.2 Překlad	76
D Ovládání programu	77
E Galerie převedených obrázků	78
E.0.3 LDR	78
E.0.4 HDR	78

Příloha A

Obsah DVD

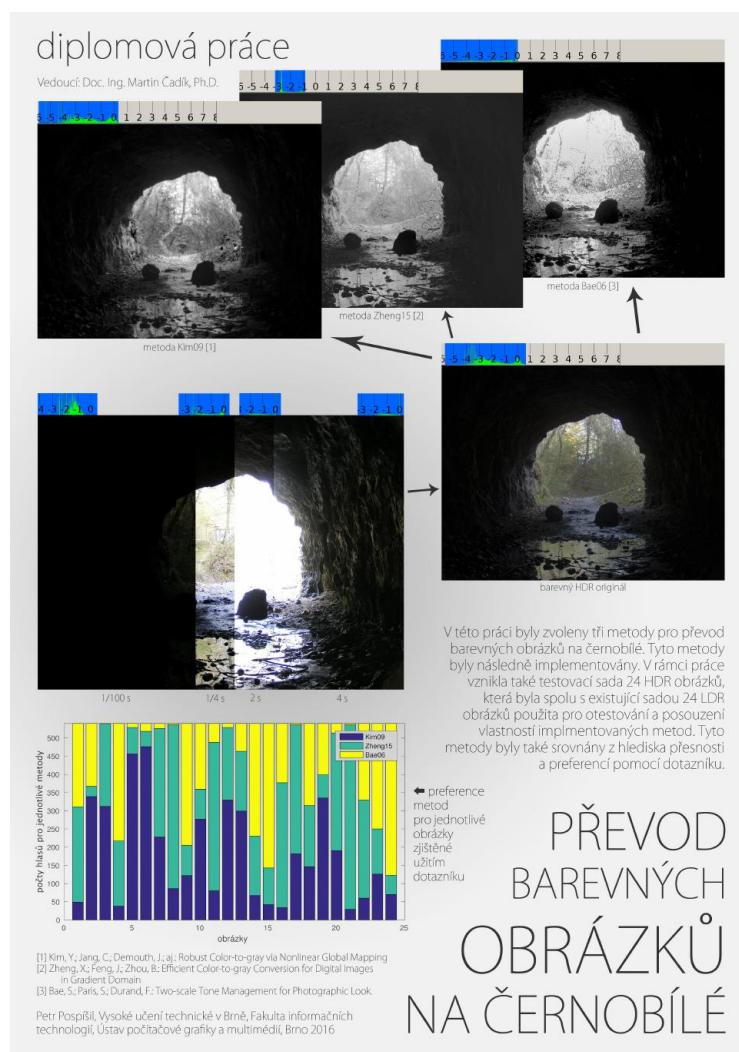
Příložené DVD obsahuje mimo jiné tyto adresáře:

DVD	
poster.....	plakát prezentující dosažené výsledky
questionnaire ..	zdrojové kódy dotazníku, včetně souboru <code>allData.sql</code> , který obsahuje všechna nasbíraná data
scripts.....	skripty použité při manipulaci s testovací sadou
src	zdrojové kódy
TMS.....	obraz git repozitáře
TMOKim09.....	zdrojové kódy metody Kim09
TMOZheng15.....	zdrojové kódy metody Zheng15
TMOBae06	zdrojové kódy metody Bae06
: ostatní adresáře projektu TMS, např. <code>TMOCmd</code> nebo <code>tmolib</code> ; obsah těchto adresářů většinou nebyl v rámci tohoto projektu měněn
srcMatlab.....	zdrojové kódy pro MATLAB ke grafům použitým v této práci
testSet.....	testovací sada obrázků
HDR	HDR obrázky
bush	soubory obrázku bush; obsahuje barevný originál ve formátu EXR a dále obrázky převedené všemi metodami a se všemi modely (pro Bae06) ve formátech EXR a TIFF
src	zdrojové fotografie pro tvorbu HDR obrázku
srcOriginal	zdrojové fotografie v originálním rozlišení
: obdobně pro ostatní obrázky
LDR.....	LDR obrázky
1.....	soubory obrázku 1; obsahují originál ve formátu TIFF a obrázky převedené všemi metodami a se všemi modely (pro Bae06) ve formátech EXR a TIFF
: obdobně pro ostatní obrázky
models.....	modelové obrázky
thesis.....	zdrojové soubory textu práce, obsahuje mimo jiné soubor <code>projekt.pdf</code> – text práce ve formátu PDF
images.....	obrázky použité v textu práce

Příloha B

Plakát

Plakát byl navržen pro tisk na papír formátu A2 a zde se nachází pouze jeho zmenšenina. Plakát v plné velikosti je k dispozici v na DVD (A), ve složce poster.



Obrázek B.1: Plakát prezentující výsledky práce.

Příloha C

Překlad a instalace programu

Pro získání informací o překladu lze také použít soubor `README.txt` umístěný na DVD [A](#) ve složce `src/TMS/`.

C.0.1 Závislosti

Soubory `matrix.h` a `pfstmo.h` byly vloženy přímo do adresáře `tmolib`, není tedy třeba se nimi více zabývat. To stejné platí i pro soubory `laplace.cpp` a `laplace.h` umístěné v adresáři `tmolib/poisson_pde`.

Pro úspěšný překlad je ale třeba mít nainstalované následující knihovny v těchto verzích nebo novější:

- OpenExr 1.2.2-4.4
- tiff4 3.8.2-8
- jpeg62 6b-14
- Boost.MultiArray 1.61.0
- FFTW 3.34

C.0.2 Překlad

Překlad se skládá ze dvou kroků. Projekt TMS využívá pro správu Makefilů nástroj `cmake`. Prvním krokem je vytvoření Makefilů. To se provede spuštěním programu `cmake` v adresáři TMS s parametrem aktuálního adresáře, tedy takto:

```
cmake .
```

Druhým krokem je samotný překlad a slinkování. Toho se docílí spuštěním programu `make`, také v adresáři TMS:

```
make
```


Příloha D

Ovládání programu

Po překladu programu dojde k vytvoření dvou spustitelných souborů: `tmocmd` a `tmogui`. Druhý jmenovaný nebyl při řešení tohoto projektu použit. Program `tmocmd` má následující použití:

```
./tmocmd method [-arg1 -arg2 ... -argn] filename
```

Program při jednom spuštění převede právě jeden obrázek (specifikovaný parametrem `filename`) a pak se ukončí. Při spuštění bez parametrů dojde k výpisu dostupných (přeložených) metod včetně jejich parametrů a také k výpisu nápovědy.

Např. převod obrázku `test.tiff` metodou `Kim09` bez specifikace parametrů se provede takto:

```
./tmocmd Kim09 test.tiff
```

Příloha E

Galerie převedených obrázků





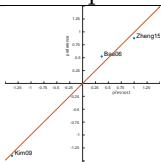
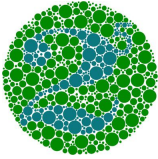
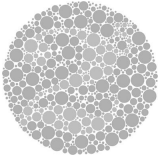
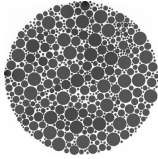
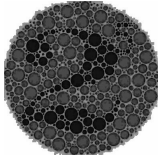
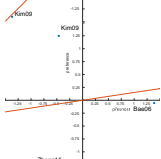




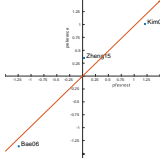

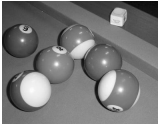
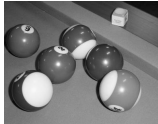

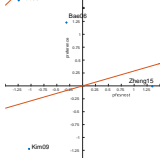




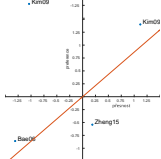



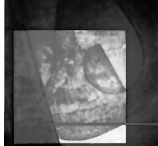
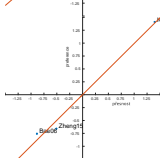




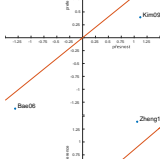
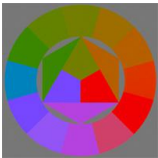

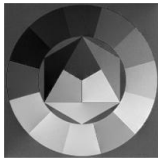
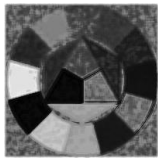
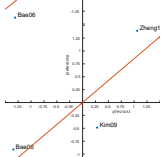

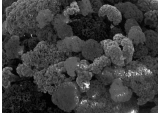
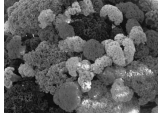
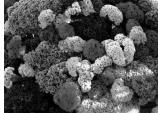
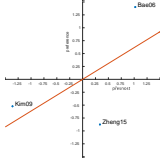




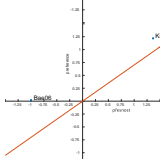




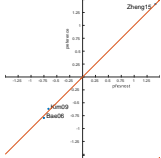
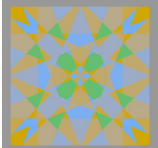

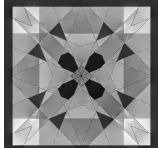

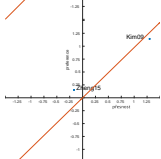
V této kapitole jsou umístěny testovací sady LDR a HDR obrázků. Pro podrobné studium výsledků je vhodné obrázky otevřít z přiloženého DVD ([A](#)), protože do tohoto textu byly z ekologických důvodů vloženy s vysokým DPI a navíc HDR obrázky jsou zde zobrazeny s omezeným rozsahem.

E.0.3 LDR

Přehled celé testovací sady LDR obrázků se nachází v tabulkách [E.1](#) a [E.2](#). V tabulce jsou zobrazeny originální obrázky, obrázky převedené všemi metodami a v posledním sloupci také graf přesnosti a preferencí pro daný obrázek.

E.0.4 HDR



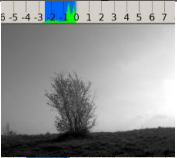
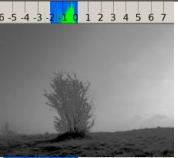




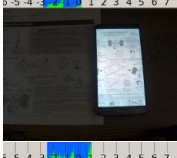
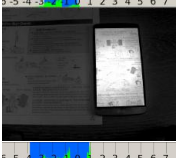


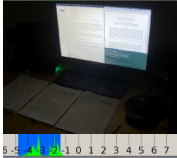
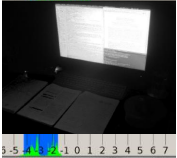
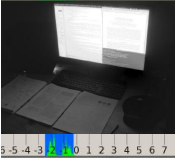
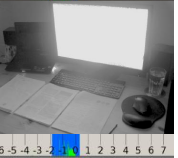
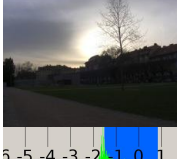
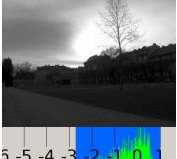
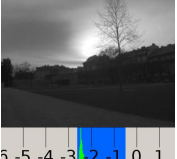

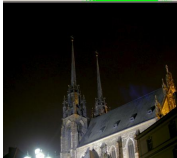
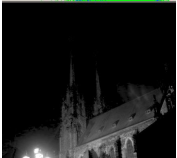
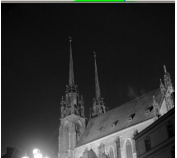
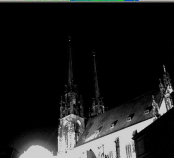
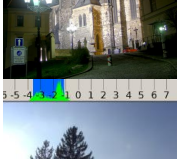
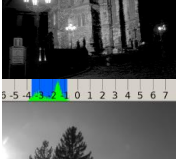
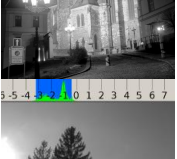
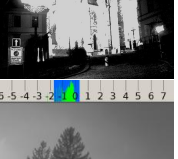


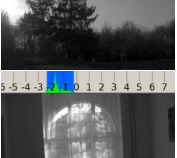
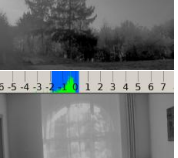


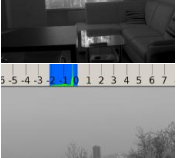
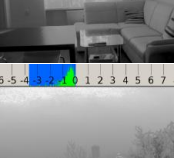



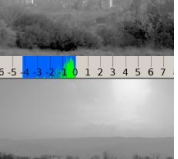

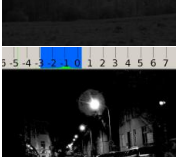

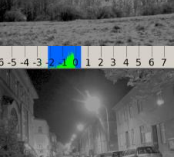



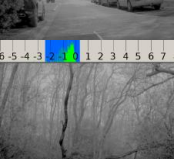
HDR testovací sada se nachází v tabulkách [E.3](#) a [E.4](#).

id	originál	Kim09	Zheng15	Bae06	přesnost a preference
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					

Tabulka E.1: Převězené testovací LDR obrázky spolu s originály, část 1 – 12/24

id	original	Kim09	Zheng15	Bae06	přesnost a preference
13					
14					
15					
16					
17					
18					
19					
20					
21	<p>The histogram generated by a physical device is generally more accurate than the histogram generated by a computer. The histogram generated by a computer is generally less accurate than the histogram generated by a physical device. The histogram generated by a physical device is generally more accurate than the histogram generated by a computer. The histogram generated by a computer is generally less accurate than the histogram generated by a physical device.</p>	<p>The histogram generated by a physical device is generally more accurate than the histogram generated by a computer. The histogram generated by a computer is generally less accurate than the histogram generated by a physical device. The histogram generated by a physical device is generally more accurate than the histogram generated by a computer. The histogram generated by a computer is generally less accurate than the histogram generated by a physical device.</p>	<p>The histogram generated by a physical device is generally more accurate than the histogram generated by a computer. The histogram generated by a computer is generally less accurate than the histogram generated by a physical device. The histogram generated by a physical device is generally more accurate than the histogram generated by a computer. The histogram generated by a computer is generally less accurate than the histogram generated by a physical device.</p>	<p>The histogram generated by a physical device is generally more accurate than the histogram generated by a computer. The histogram generated by a computer is generally less accurate than the histogram generated by a physical device. The histogram generated by a physical device is generally more accurate than the histogram generated by a computer. The histogram generated by a computer is generally less accurate than the histogram generated by a physical device.</p>	
22					
23					
24					

Tabulka E.2: Převěně testovací LDR obrázky spolu s originály, část 13 – 24/24

název	originál	Kim09	Zheng15	Bae06
bush				
cave				
mobile				
monitor				
park				
petrov				
tree				
window				
feec				
landscape2				
slovinska				
forest				

Tabulka E.3: Převedené testovací HDR obrázky spolu s originály, část 1 – 12/24

název	originál	Kim09	Zheng15	Bae06
lisen2				
stump				
hadyTower				
kitchen				
lamp				
landscape				
landscape4				
malomerice				
malomerice2				
noveSady				
park3				
window2				

Tabulka E.4: Převedené testovací HDR obrázky spolu s originály, část 13 – 24/24